Goldbug

Goldbug - Secure Email Client & Instant Messaging User Manual

Figure 1: GOLDBUG-Logo

Secure Email Client & Instant Messaging http://goldbug.sf.net

Contents

What is Goldbug?

GoldBug is a secure email client and instant messenger.

With the use of GoldBug (GB) you can be sure, that no unwanted third party can eavesdrop on your conversations. Private user-to-user communication remains private, in protected space. For that, GoldBug uses strong multiple encryption, also called hybrid encryption, with different levels of modern encryption technologies from established encryption libraries - as libgcrypt (known from GnuPG) and OpenSSL.

For example, more than 8 public / private keys are generated for encryption - based on the RSA encryption algorithm, or optionally ElGamal and NTRU. Furthermore, the application also offers decentralized and encrypted e-mail and decentralized public E*IRC group chat. As in every messenger application as well files can be shared and sent as attachments. With the tools "Rosetta CryptoPad" and the "File Encryptor" you can securely encrypt text and/or files.

Goldbug is relaying on the code of Spot-On (http://spot-on.sf.net). Spot-on defines itself as an exploratory research project investigating a variety of communication and cryptographic algorithms. The software is composed of two separate applications, a multi-threaded kernel and a user interface. The two components are written in C++ and require the Qt framework as well as an assortment of libraries. Qt versions 4.8.x and Qt 5.x are supported. The application is available on FreeBSD, Linux, OS X, OS/2, and Windows. Please note that the Echo algorithm and its name are not based on Ernest J. H. Chang's 1982 Echo Algorithms: Depth Parallel Operations on General Graphs paper.

Why is it important that you encrypt your communications?

Currently, almost all wireless WIFI networks are protected with a password. In a few years, plain text messages or e-mails to friends over the Internet are to be encrypted as well.

This is not a question of whether you have something to hide or not, there is the question of whether we even control our own communication - or if it is controlled by others, third parties. It is ultimately a question of the attack on the free thinking and cancellation of adopting a “presumption of innocence”. Democracy requires thought and discussion of alternatives in private and in public. Strong multi-encryption (so-called "hybrid encryption") ensures the declarations of human rights in broad constituted consensus and is a digital self defense, everyone should learn and use.

The GoldBug Messenger strives to be an easy to use tool for this claim. Similarly to the security development in the automobile also the e-mail and messaging encryption will develop: we first moved the car without a seat belt, today we deal with seat belts and airbags in addition or in the third with additional safety information systems. Hence for internt communication: The unencrypted plain text email or instant message is obsolete.

Where does the name "Goldbug" derive from?

The GoldBug is a short story by Edgar Allan Poe: "In the plot it is about William LeGrand, who recently discovered a gold-colored ladybug. His buddy, Jupiter, now fears that LeGrand is obsessed to reach wealth, knowledge and wisdom - after he has been with the Gold Bug in touch; and therefore goes to another friend of LeGrand, an unnamed narrator, who agrees to visit his old friend. After LeGrand has then found a secret message and he was able to successfully decrypt it, the three start an adventure as a team.

The GoldBug story - one of the few pieces in the literature - integrats encryption texts as an element of the story. Poe was thus the popularity of encryption texts ahead of his time when he wrote "The Gold-Bug" in 1843, in which the success of the story, for example, turned to such a cryptogram and metaphorically to the search for the philosopher's stone. From history, the Gold-Bug was an instant read success and was extremely popular, and for the writer, the most studied work of Poe during his lifetime. His ideas helped to write encrypted texts and also to make the so-called cryptograms well known" (compare also Wikipedia). 170 years later, encryption has more weight than ever. Encryption should be a standard when we send communication over the insecure internet.

Encryption: GOLDBUG provides alternatives to RSA

GoldBug Messenger has several alternatives to RSA: if this encryption algorithm would once become insecure (e.g. with quantum computers). So far, RSA applies - with correspondingly large size of the key it continues to be safe. In addition to RSA GoldBug has implemented the ElGamal encryption algorithms and also NTRU.

In the signature process there are also optionally available encryption methods: So there remains a greater choice for the end-user: DSA, ECDSA, EdDSA, ElGamal and RSA. Of course, each user can set his individual key size, the "cipher", the "hash type", also "iteration count", and the salt-length - often typical criteria used for creating keys for encryption. The advantage is that each user can individually define this for himselves.

Figure 2: Alternatives to RSA

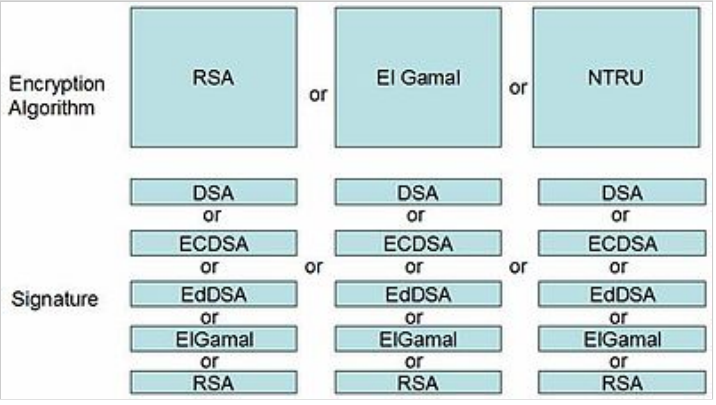


Figure 2: GoldBug provides alternatives to RSA

RSA - ElGamal and NTRU compared

NTRU is an asymmetric encryption method that was developed in 1996 by the mathematicians Jeffrey Hoffstein, Jill Pipher and Joseph Silverman. It is loosely based on lattice problems. NTRU is not known to be vulnerable to quantum computer based attacks. However NTRUEncrypt has not as well studied as more common methods (e.g. RSA). NTRUEncrypt by IEEE P1363.1 is standardized (see https://en.wikipedia.org/wiki/NTRU).

RSA (according to the persons Rivest, Shamir and Adleman) is an asymmetric cryptographic method that may be used for both, encryption and digital signature. It uses a pair of keys consisting of a private key that is used to decrypt or sign data, and a public key. The private key is kept secret and can only be calculated with extremely high effort from the public key (see https://en.wikipedia.org/wiki/RSA_(cryptosystem)).

The ElGamal encryption method or ElGamal cryptosystem is a system developed by cryptologists Taher ElGamal in 1985. It is a public-key encryption scheme which is based on the idea of the Diffie-Hellman key exchange. The ElGamal encryption scheme is based, as well as the Diffie-Hellman protocol, to operations in a cyclic group of finite order. The ElGamal encryption method is provable IND-CPA secure under the assumption that the decisional Diffie-Hellman problem is not trivial in the underlying group. Related to the encryption methods described herein (but not identical to this) is the ElGamal signature scheme (the ElGamal signature method is not implemented in GoldBug). ElGamal is not subject to patent (see https://en.wikipedia.org/wiki/ElGamal_encryption).

Block Cipher Modes of Operation

GoldBug uses CBC with CTS to provide confidentiality. The file encryption mechanism supports the GCM algorithm without the authenticity property that's provided by the algorithm. To provide authenticity, the application uses the encrypted-then-MAC (EtM) approach. The Encrypted and Authenticated Containers section provides more details.

Hybrid Encryption System

GoldBug implements a hybrid system for authenticity and confidentiality. One portion of the system generates per-message authentication and encryption keys. These two keys are used for authenticating and encapsulating data. The two keys are then encapsulated via the public-key portion of the system. The application also provides a mechanism for distributing session-like keys for data encapsulation. Again, the keys are encapsulated via the public-key system. An additional mechanism allows the distribution of session-like keys via previously-established private keys. Digital signatures are optionally applied to the data. As an example, please consider the following message: EPublic Key(Encryption Key || Hash Key) || EEncryption Key(Data) || HHash Key (EEncryption Key(Data)). The private-key authentication and encryption mechanism is identical to the procedure discussed in the Encrypted and Authenticated Containers section.

Public Key Infrastructure

GoldBug utilizes the libgcrypt and libntru libraries for permanent private and public key pairs. Presently, the application generates ten key pairs during the initialization process. Key generation is optional. Consequently, GoldBug does not require a public key infrastructure. ElGamal, NTRU, and RSA encryption algorithms are supported. DSA, ECDSA, EdDSA, ElGamal, and RSA signature algorithms are supported. The OAEP and PSS schemes are used with RSA encryption and RSA signing, respectively. Communications between nodes having diverse key types are well-defined if the nodes share common libgcrypt and libntru libraries. Non-NTRU private keys are evaluated for correctness via the gcry_pk_testkey() function. Public keys must also meet some basic criteria such as including the public-key identifier.

What is the echo protocol?

With the Echo Protocol or sometimes Echo-System is meant - simply expressed - that

• each message transmission is encrypted ...

• ... and each network connection node sends each message to each connected neighbors. And so on.

That´s it.

```
Example for the Echo encryption:

TLS/SSL (AES (RSA* (Message)))
```

-) Instead of RSA ElGamal or NTRU may also be used,

First, you write the message. It is enrypted in an asmymmetric way and uses the private/public key infrastructure of the chosen encryption alogorithm, e.g. RSA. The plaintext message is hashed, and the hash plus the encrypted messages are wrapped into one transmission. If the receiver is able to decrypt the pciphertext to plaintext, and the hashed plaintext is the same hash of the provided hash, the messages was decoded successfully and displayed to the user. If not, the ciphertext of the message and the hash of the plaintext-message are sent along to all neighbors - as we assume it is a message for other users.

Further, the messages is then sent through an established channel, based on symmetric encryption, if you provide a "call" and set a "gemini" (see below for AES). Third, all that, the asymmetric encryption, the symmetric encryption is sent through an (decentral and self signed) TLS/SSL Channel to the other user.

Thus, GoldBug implements the Echo Protocol of Spot-on. The Echo is a malleable protocol. That is, the protocol does not require rigid implementation details. Each model may adhere to their own peculiar obligations. The Echo functions on the elementary persuasion that information is dispersed over multiple or singular passages and channel endpoints evaluate the suitability of the received data. Because data may become intolerable, GoldBug implements its own congestion control algorithm. Received messages that meet some basic criteria (hashed values) are labeled and duplicates are discarded. Advanced models may define more sophisticated congestion-avoidance algorithms based upon their interpretations of the Echo.

GoldBug provides two modes of operation for the general Echo Protocol: Full Echo and Half Echo. The Full Echo permits absolute data flow. The Half Echo defines an agreement between two endpoints. Within this agreement, information from other endpoints is prohibited from traveling along the private channel.

Full Echo

It is based on the so-called "small world phenomen": Anyone can reach everyone somehow over seven corners or hops in a peer-to-peer or friend-to-friend network - or simply a circle of friends can be reached over an installed and shared echo chat server.

Half Echo

The mode of the "Half echo" sends a message as a single hop, i.e. for example from Bob to Alice. Alice sends the message then no longer forward (as it is the default in the pure/full echo).

In addition to full echo, and half echo there is, third, the Adaptive Echo (AE). Here, the message will be sent only to neighbors or friends, if they know a cryptographic token, so they must have it previously shared and saved. Who does not know the token, does not get the message forwarded. The chapter below about Adaptive Echo (AE) reports in more detail about this option.

Echo Accounts

Finally, the echo still knows echo accounts. A type of firewall. This can be used to ensure that only friends, who know the account access, can connect. Thus, a Web-of-Trust is created, i.e. a network exclusively among friends. It is not based on the key for encryption, it is independent of it. That means you do not have to even associate your public key with your IP address or even announce your IP to the network of friends, to a DHT where users can search for it.

Figure 3: Simulation of the echo-network

Basically, in echo each node sends the message to each node. If you do not get a message a second time, it has been in a temporary memory compared to previous gotten messages (using the hash value for this message) and possibly rejected, if a doublette has been recognized ("Congestion Control").

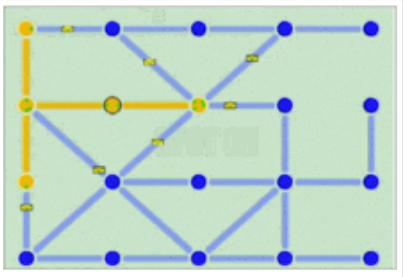


Figure 3: Graphical simulation of the echo network: Every node is sending the message to every connected node.

Finally, you can also send out spurious messages ("fake messages") and simulated communication messages ("impersonated messages") with the application GoldBug. In these messages the encryption is no encryption, but represents pure random characters that are sent out from time to time, and for the other case a human conversation is simulated based only on just random characters too, which perform a kind of chat conversation from human beings. Thus, the analysis of messages can be more difficult, if third party would record messages (as a "recorder") – and remember that this must be possibly accepted, that all your communications will be stored and recorded.

Further, GoldBug implements a plain, and perhaps original, two-pass mutual authentication protocol. The implementation is well-defined with or without SSL/TLS. The protocol is weakened if SSL/TLS is neglected, however.

The Accounts procedure is as follows:

1. Binding endpoints are responsible for defining account information. During the account-creation process, an account may be designated for one-time use. Account names and account passwords each require at least 32 bytes of data.

2. After a network connection is established, a binding endpoint notifies the peer with an authentication request. The binding endpoint will terminate the connection if the peer has not identified itself within a fifteen-second window.

3. After receiving the authentication request, the peer responds to the binding endpoint. The peer submits the following information: HHash Key(Salt || Time) || Salt, where the Hash Key is a concatenation of the account name and the account password. The SHA-512 hash algorithm is presently used to generate the hash output. The Time variable has a resolution of minutes. The peer retains the salt value.

4. The binding endpoint receives the peer's information. Subsequently, it computes HHash Key(Salt || Time) for all of the accounts that it possesses. If it does not discover an account, it increments Time by one minute and performs an additional search. If an account is discovered, the binding endpoint creates a message similar to the message created by the peer in the

previous step and submits the information to the peer. The authenticated information is recorded. After a period of approximately 120 seconds, the information is destroyed.

5. The peer receives the binding endpoint's information and performs a similar validation process, including the analysis of the binding endpoint's salt. The two salt values must be distinct. The peer will terminate the connection if the binding endpoint has not identified itself within a fifteen-second window. Please note that the Accounts system can be promoted by including an encryption key. The additional key will allow for finer time resolutions.

If SSL/TLS is not available, the protocol may be exploited. A relay station may record the values in the 3rd step and subsequently provide the information to the binding endpoint. The binding endpoint will therefore trust the foreign connection. The recording device may then seize the binding endpoint's response, the values in the 4th step, and provide the information to the peer. If the information is accurate, the peer will accept the binding endpoint's response.

The ECHO Grid

When students speak and teach about the echo protocol, then we simply draw an ECHO grid with the letters E_C_H_O and number the nodes of E1 to O4 and connect the letters with a connecting line on the ground. For example, the connection E1-E2 then identifies an IP connection to a neighbor.

Figure 4: The ECHO Grid

Figure 4: The ECHO Grid: To teach about the echo protocol just use the characters E_C_H_O and build an ECHO-Grid-template.

If the individual accounts now point to exchanged keys (instead of IPs) – then a new layer on top of the level of IP connectivity of a P2P / F2F network is produced.

Figure 5: Alice, Bob, Ed and Mary in the ECHO Grid

Examples of key-exchange of Alice, Bob, Ed and Mary.

• Alice (IP = E1) and Bob (IP = C3) exchanged their public keys and are connected via

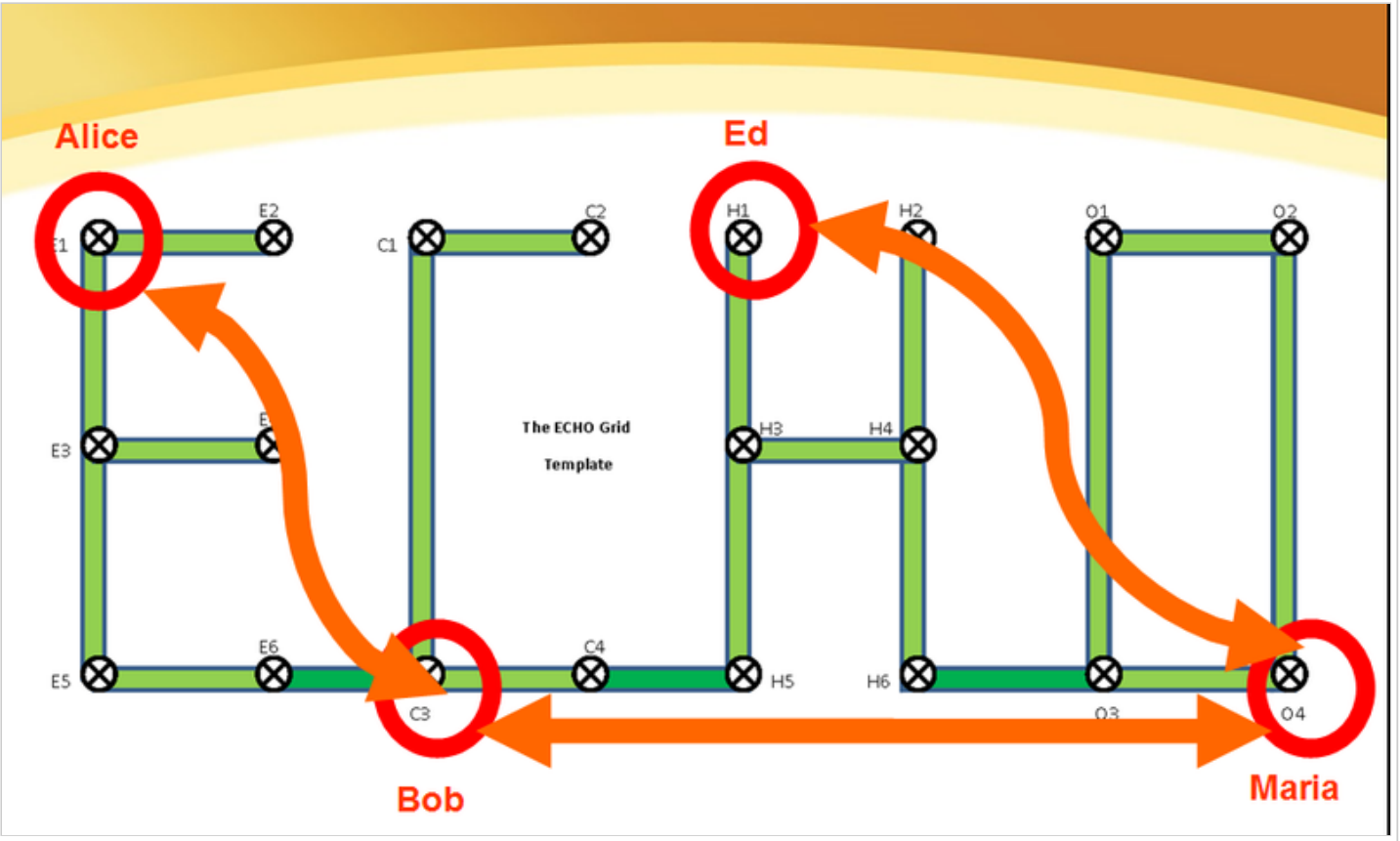


Figure 5: Alice, Bob, Ed and Mary in the ECHO Grid - Example of the ECHO.

the following IP Neighbors: E1-E3-E5-E6-C3.

• Bob (C3) and Maria (O4) are just friends, they have changed their public key for encryption as well: and use the IP connections of neighbors: C3-C4 H5 H3-H4-H6-O3-O4.

• Finally, Maria (O4) is a friend of Ed (H1). Communicate via either the way: O4-O3-H6-H3 H4-H1 or use the path of O4-O2-O1-O3-H6-H3 H4-H1. Since the echo protocol sends each message to each connected IP neighbor, the path will be successful, to deliver the message to any IP neighbor, which is the fastest.

• Direct IP connections from neighbors such as E1-E3 can by creating a so-called "Echo accounts" are hedged. No other IP address can connect to the so-called "listener" neighbor as E3 to listener E1. Using this method, a web-of-trust will be created - without being dependent on encryption keys - still you need a friend with whom you are trading your chat or e-mail key.

• So-called "Turtle hopping" is much more efficient in an Echo Network: When Ed and Alice exchange a so-called "StarBeam Magnet" for file transfer, then the echo protocol transports packets on the path H1-H3-H5-C4-C3-E6 E5-E3-E1. Mary is not in the route, but you will get the packages as well over the full echo when she knows the StarBeam Magnet. Advantage is that the hopping does not go over the key, but over the IP connections (e.g. the Web of Trust). Basically everything is always encrypted, so why not take the shortest route?

• A so-called "Buzz" and "echo-ed IRC Channel" (E*IRC)-room can e.g. be create or "hosted" by the nearest node O2. Since only the referring user knows the Buzz-Room name, all other neighbors and friends are left out. Benefit: In this example you can talk with unknown friends in one room without using a public-RSA-Key – or to have ever exchanged asymmetric keys. Instead, you can simply use a single-magnetic ("one-time-magnet") for a "buzz" / "E*IRC" room.

• Maria is a mutual friend of Ed and Bob and activates the C/O (care of) function for emails: This allows Ed, to send E-mail to Bob even when he is offline, because: Maria saves the e-mails in her cache until Bob then comes online.

• Furthermore: Alice created a so-called virtual "Email Institution". This is not comparable to a POP3 or IMAP server because the e-mails are only cached: Ed sends his public email key to Alice - and Ed adds the magnets of the "Email institution" by Alice within his program. Now the emails from Bob and Ed are cached at Alice (in the e-mail Institution), even if Maria should be offline.

It is helpful to follow the examples in the graph above.

Adaptive Echo (AE) and its AE-tokens

For the explanation of the "adaptive echo" another echo-grid can be drawn with the related points A and E.

Figure 6: The "Hansel and Gretel" - Example of adaptive echo

If you, your chat friend and a created third node point as a chat server insert in the program the same AE token ("Adaptive Echo token"), the chat server will send your message only to your friend - and not to all other connected neighbors or users as it would normally be the case within the full echo mode. With an AE token, no one else will receive your message or can see, that you communicate. So therefore possible neighbors, and potential "recorders" will be excluded, to be able to record any communications and then want to try to break the multiple encryption to come to the message kernel inside the several layers of encryption.

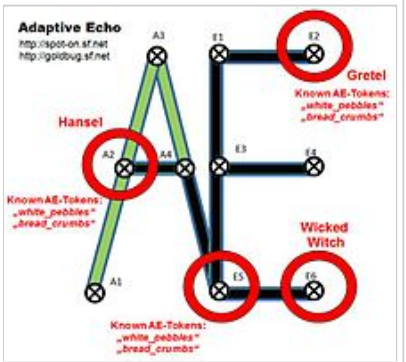


Figure 6: The "Hansel and Gretel" - Example of the Adaptive Echo: Cryptographic Tokens provide access to messages and graphs or not. A learning, adaptive network.

The Adaptive Echo is a complement to the Echo Protocol and substantiates the opinion that the Echo Protocol is a malleable method. Endpoints that bind multiple parties may optionally define Adaptive Echo tokens. Adaptive Echo tokens are composed of authentication and encryption keys as well as details about the choice algorithms. If configured, binding endpoints are able to permit or restrict information travel based on the content of the data. As an example, peers that are cognizant of a specific Adaptive Echo token will receive data from other cognizant peers whereas traditional peers will not. Binding endpoints therefore selectively-echo data.

The Adaptive Echo behaves as follows:

1. A binding endpoint defines an Adaptive Echo token. The information must be distributed securely.

2. A networked peer having the given Adaptive Echo token generates HHash Key(EEncryption Key(Message || Time)) || EEncryption Key(Message || Time) where the Encryption Key and Hash Key are derived from the Adaptive Echo token. The generated information is then submitted to the binding endpoint as Message || Adaptive Echo Information.

3. The binding endpoint processes the received message to determine if the message is tagged with a known Adaptive Echo token. If the message is indeed tagged correctly, the Time value is inspected. If the Time value is within five seconds of the binding endpoint's local time, the message is considered correct and the peer's presence is recorded.

4. As the binding endpoint receives messages from other peers, it inspects the messages to determine if the messages have been tagged with Adaptive Echo tokens. This process creates a network of associated peers. Because peers themselves may be binding endpoints, the Adaptive Echo may be used to generate an artificial trust network.

Adaptive Echo is susceptible to eavesdropping. As an example, if a message that is tagged with an Adaptive Echo token should travel through one or more peers to reach a destination, the peers may record the message and subsequently replay the message to a binding peer. The replay must occur within the acceptance window of the message. Additionally, the binding endpoint's congestion control container must not already contain the message. If both conditions are met, the binding endpoint will consider the peer as trustworthy.

Hansel and Gretel - an example of the Adaptive Echo mode:

If node A2, E2 and E5 use the same AE token, then point E6 accounts will not receive a message that the node A2 (Hansel) and the node E2 (Gretel) exchange. After all, the node E5 learns about the known token "White pebbles" no to send messages to the node in point E6: the "Wicked Witch". It is a learning or adaptive network.

An "adaptive echo" network thereby reveals no target information (see also " Ants routing "). Remember: the mode of "Half Echo" sends only one hop to connected neighbors and the "Full Echo" sends the encrypted message to all nodes connected via an unspecified number of hops. While "Echo Account" helps or hinders other users almost as a firewall or authorization concept in joining, however, "AE-tokens" keep graphs or paths exclusivity – and it does it also for messages, that are sent via connecting nodes, that know the AE-token.

Chat server administrators can exchange their tokens with other server administrators - if there is trust among themselves defined (so-called "ultra-peering for trust") and they want to build a web of trust based on the Adaptive Echo tokens.

In a network lab or at home with three, four hosts, you can simply try out the Adaptive Echo and repeat this settings:

Use "SPOTON_HOME" as a file in binary directory to launch multiple program instances on a single machine and connect the instances - or just use a network with three or more computers. So then follow this procedure:

1. First Create a node as a chat server.
2. Create two nodes as clients.
3. Connect the two clients to the chat server.
4. Exchange keys between the clients.
5. Test the normal communication skills among both clients.
6. Set an AE token on the server.
7. Test the normal communication skills among both clients.
8. 8 Now use the same AE token in a client.
9. 9 Write down the result: The server node stops sending the message to other nodes, which do not have the AE-token or don’t know it.

This example should be easy to be replicated.

3.6 How the ECHO protocol works

Referring now together the different methods and options, the following chart can provide a complex overview.

Figure 7: How does the ECHO PROTOCOL work?

Shown in the graph are the different usage examples of "Full Echo", "Half Echo" “Adaptive Echo" and "Echo Accounts".

A distinction is made between physical-IP-connections and “virtual-connections” to keys. Keys are therefore not necessarily associated with an IP connection!

Users can replace an asymmetric public key, and also use magnet-URIs with symmetric encryption details, as well as tokens

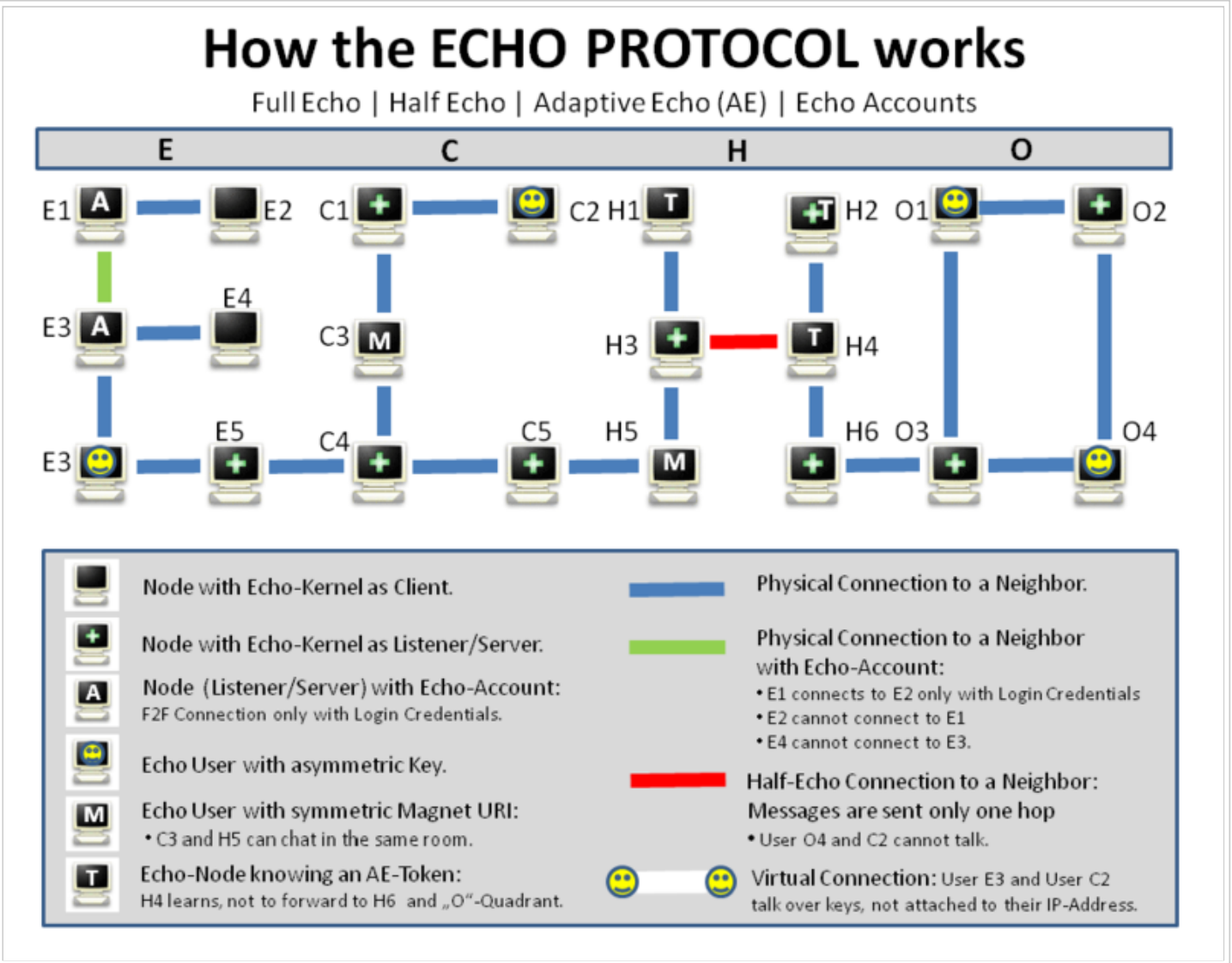


Figure 7: Graphic: How the Echo Protocol works - Examples.

and account credentials.

Connection nodes can accept and prohibit addressed connections - as well as dedicated addressed messages.

Accordingly, different communication scenarios arise.

Examples:

a. User H4 has an AE token. It does not send messages (via the connecting node H6) in the O-quadrant, when H6 does not know the token.

b. If H3 sends a message to H4, H4 then sends this message not just because it is a combination of "Half echoes".

c. The user E2 cannot connect to the user E1, since he does not know the login for the echo account.

d. Users O1 and O4 chat with each other and knowing only their public key for encryption.

e. User H3 and C5 chat via a URI magnets in the same group chat room (also known as Buzz or E*IRC – echoed IRC).

.

Screenshots: Password definition, key generation and kernel-activation

The Goldbug Messenger has an interface and a kernel. Both are given as a binary (i.e. under Windows called GoldBug.exe and Kernel.exe).

With the user interface (called “interface” or “GUI” (Graphical User Interface, GUI = GoldBug.exe)) the kernel must be activated before every start, then the direct connections to your friends or on a common chat server or the echo network are coordinated. However, before the program can be started at all, first there must be an "Initial Setup", i.e. you have to create your keys for encryption. There are currently 8 key generated, which can take on slow machines up to approximately one minute. Similarly, a passphrase for the Messenger is to choose which is requested each time as login, after you have the program, the Goldbug.exe, started. The password must be at least 16 characters long. If that is too long, you can also repeat a password three times, such as "password_password_password", but the password is then not as secure as one with a random string. When you start Goldbug the first time, add in the blue box a nickname and define a passphrase. There are two methods: the passphrase method or the Q&A (Question and Answer) method.

Figure 8: Set password-generating key - and active the kernel

The two methods can be distinguished as follows:

• Passphrase method: hash (passphrase + salt), which means a " salted hash "is used. When creating a password that is not stored locally, but only the hash of the input.

• Q/A method: hash (Question, Answer), which means a " HMAC "is used. And neither the question nor the answer is stored on your machine and no salt is generated by the machine at random. Instead of a question you can also type two passwords without the question mark, of course. Note, that here the question and the answer must be entered exactly in subsequent logins, as defined herein, and no other input check ("Password Confirmation") is given, similar to the password method above.

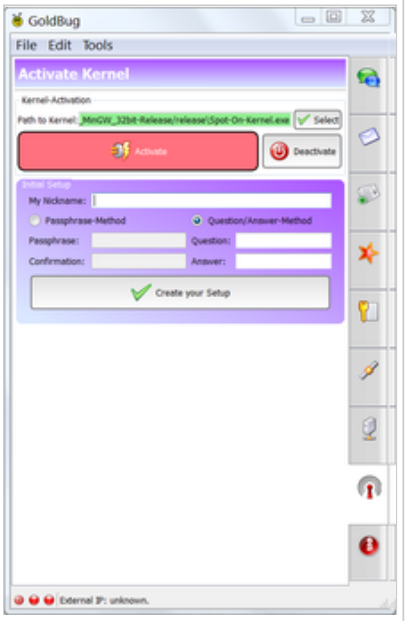


Figure 8: Set password-generating key - and active the kernel - Initial Start: Generate a password for the GoldBug Messenger

At once the keys are generated, you can enable the kernel. Press the red button to "Activate" the kernel and then make sure that the file-path is specified for kernel.exe and is thus highlighted in green. If not, change the path and pick up the kernel.exe At the initial activation of the project-chat-server's IP address is added as a neighbor automatically and this serves as a temporary chat server through which you can chat with your friends as a test until you have created your own connection node on a web server or at home for directly connections. Please use the test server of the project only for scientific or test trials. If you want to connect directly without a server, one of the users must create in the so-called “Listener Tab” a Chat-server and enable the firewall for port - and port forwarding in the router - in addition to your machine.

When you start the GOLDBUG Messenger for the first time you are asked by a pop-up window, if you want to activate the kernel. Otherwise, for all other starts you have to press the red "Activate kernel" button after login – before you can chat. If it's green, the kernel is running. If you close the GUI, the kernel will continue to run. It is therefore advisable to first disable the kernel and then close the GUI. But in any case, another pop-up window will ask you, if both are to be closed. Otherwise you are running the kernel GUI-less, which is indeed sometimes wished on a web server, so no one has access to an opened interface. You can also enable / disable the kernel by pressing the first LED in the status bar at the bottom left. If it is green, the kernel is active - when it is red, the kernel is off. Your generated keys are stored in the sub-path"/.spot-on". If you want to set up a new login with new keys and all user data should be deleted, then just delete that path and reboot. The same can be achieved in the main menu with the command: "!!! “!!!_Total_Database_Erase_!!!".

Described so far is the minimum visibility of the interface: From the main menu, you can also choose between "full view" or "minimal view". Anyone who knows not that good with computers, should choose the minimal view, as it hides a range of options, which may not be required. Keep it simple.

The non-minimal view shows in the tab "Activate kernel" the following additional elements:

• Path to the kernel: Here you can enter the kernel path. If in the path to the kernel with the "spot-on-kernel.exe" is specified correctly, then the path is highlighted in green. Otherwise, look where the executable file of the kernel is or copy it as well to the executable file of the GUI (goldbug.exe) or adapt the path accordingly.

• PID: The PID number identifies the process ID with which the executable is characterized in Windows. You can find the same process ID in the Windows Task Manager.

• Simulacra: This function sends upon activation of the check box a "simulated" chat message to the Echo Network. This "fake" message consists of purely random numbers and makes it harder for analysts, to distinguish encrypted messages with real and random messages. Simulacra is a term that is not unknown from both, the movie "The Matrix" and in the philosophy of Baudrillard.

• Impersonator: In addition to random cipher-text, also fake messages can be simulated, as if a real person chats from time to time and sends responses to a chat from the GOLDBUG program to another participant. These messages are filled with pure random data, but vary – as simulated in a real chat conversation.

• Create Settings: For the key generation you should select a key larger than 2048 bits and you can even choose other options such as algorithm, hash type, cipher, salt-length or iteration count.

• With the "regeneration" function you can also regenerate individual keys - with new values and options. By checking out the check box, set the values and re-generating the respective keys. But then you must exchange your new key again with your friends, because the key is your communication ID.

Just swap your key with a friend - and a first chat can begin! Set the key exchange as follows:

Start a first secure chat

You can find after a successful key exchange your chat friend in the tab "Chat".

Add contact by exchanging a key

As a friend is added and the key is exchanged, has already been discussed above. After the connection to a chat server has been explained in the previous section, you are to begin with two green LED lamps in the status bar and a friend in chat tab normally be able to chat. If this is not the case, check if the two friends use the same version of the program. Then it may be a matter for the advanced user, sometimes debug a private chat server or connect via a direct connection from home to home and also to define their own routers for home Internet connection.

You and your partner, two friends, each must exchange their public keys. First copy out the key and then paste the key of your friend in the tab "Add Friend" ("Add Friend / Key") and press the button enter.

Your friend can send the key by e-mail or via another chat program. Then copy it into this tab and press the "Add" button at the bottom.

You can find your own key as well in the tab "add friends" ("Add Friend / key"). About the big button ("Copy Keys") above you can copy-out your key to the clipboard.

Goldbug uses a public / private key infrastructure, as it is also known, for example, from GnuPG. The public key can be exchanged, and the private key is encrypted on your hard drive.

The different functions of Goldbug have accordingly for security reasons different key pairs. For Email a different key is used than for the chat. But there are in the copy-out button the function to copy out all the keys in a single text (“Overall-key”). Copy here the full text and send this to your friend.

Your friend does the same and you're adding the friend’s key in the text box. (If necessary, it may be necessary to confirm with the right mouse button in the context menu a new friend as a friend (Make-Friend). This will most often be used when a friend sends his key online in a direct IP connection (which is possible too). This function is given in the interface of spot-on – but in the user interface Goldbug this is not available, so that always both participants copy and paste their keys. But if a friend uses the spot-on client here and builds a direct IP connection to a user of the Goldbug client, then it would be theoretically possible to transfer the key also via IP connection instead of copy / paste).

Finally - after key exchange - the friend appears with his nick name in the chat tab or email tab.

Key derivation

GoldBug uses separate authentication and encryption keys for local data. The key-derivation process is as follows:

1. Generate a cryptographic salt. The size of the salt is configurable.

2. Derive a temporary key via the PBKDF2 function. The hash algorithm, iteration count, passphrase (question/answer), and salt are input parameters to the function. All of the aforementioned parameters are configurable.

3. Using the temporary key from the previous step, derive a new key via the PBKDF2 function. The previous parameters are also used, however, the temporary key replaces the passphrase (question/answer).

4. Separate the derived key into two distinct keys. The encryption key is N bytes long, where N is the recommended key size of the selected cipher. The remaining bytes compose the authentication key. The generated authentication key contains at least 512 bytes.

Special feature: Repleo

If you have already received a key of your friend and have inserted it, but now your public key should not be exposed, you do not want it to be know or to be stored in an e-mail program, then you can encrypt your own key with the obtained key of your friend: This is called REPLEO.

When you send a Repleo, your public key is already encrypted with the public key of your friend.

Your friends can also copy and paste the Repleo into the tab "Add Friend / Key" – just change the radio button to Repleo.

A key always starts with a letter "K" or "k" and a Repleo starts always with an "R" or "r". So you can determine whether it is a Key or a Repleo to the corresponding textbox with two radio buttons.

Figure 09: Tab Key: insert key and confirm with the add-button

Starting the first chat

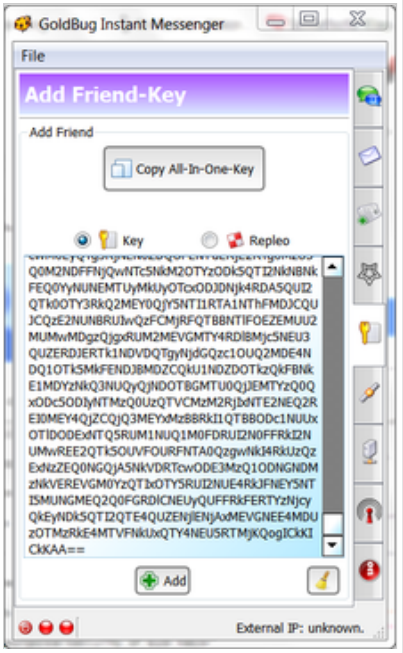


Figure 09: Tab Key: Exchange a Key or Repleo with a friend and add it to the Key-Tab.

To be able to chat, both participants should ideally use the same and the latest version of the program, have their keys generated and exchanged and be connected on the web to a network node or chat server. If the first two LEDs in the status bar at the bottom light green and the names of your friend appear in chat tab below, it already looks good.

Figure 10: Chat tab

If the online status of your friend is blue (absent), red (busy) or green (ready to talk), the chat can begin. Either select the friend in the participants table and chat out of the chat Tab, or double-click with the mouse on the desired friend and a pop-up chat window for that dedicated friend opens.

Chatting in a pop-up window

Figure 11: Start a pop-up chat window with a double klick

The advantage to chat in the chat tab is, that you can select multiple friends so that the same message reaches all friends. If you use the pop-up chat then you no longer have to pay attention to the highlighting of the right friend in chat tab: Messages in the pop-up window are only sent to one dedicated friend exclusively.

Chat-Emoticons

Goldbug uses an entire bouquet of emoticons - also called smileys. To use the help, double-click on a friend, so that a pop-up chat window for private chat. Now go with the mouse over the send button. In a tooltip that appears smileys are displayed and the input of the ASCI codes for emoticons will be displayed in the chat. In the chat/options-Tab is also the ability to enable & disable the graphical representation of smileys.

Figure 12: List of Emoticons

Additional Security Feature: MELODICA

MELODICA stands for "Multi Encryted LOng DIstance Calling". It refers to call a friend like with a phone - only that a secure end-to-end encryption is enable and built.

The end-to-end passphrase - also called Gemini - should remain a secret between two parties. Therefore, the electronic transmission is always a problem when it can potentially be intercepted. Goldbug has this transmission problem solved by the Gemini which is transmitted with a symmetric encryption by a further encrypted channel. Gemini is the term for twins, i.e. it refers to both parties, which should know the passphrase, technically an end-to-end encryption is generated.

Figure 13: The icon of the MELODICA Button

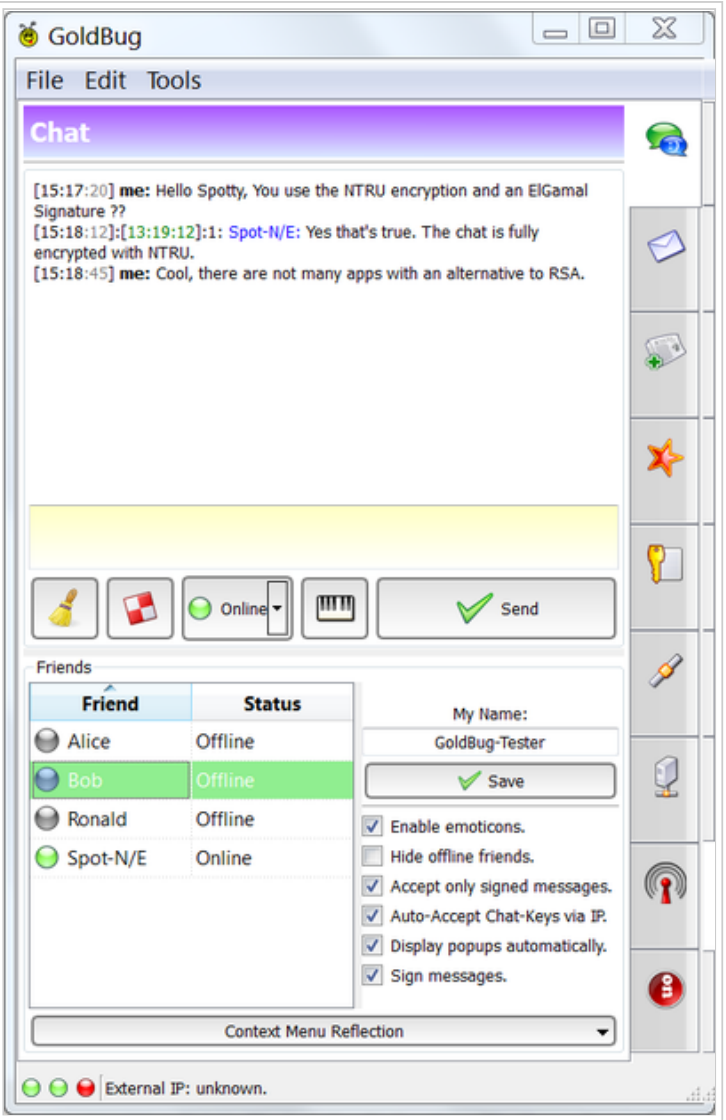


Figure 10: Chat-Tab of GoldBug-Messenger

The MELODICA button is creating a "Call", a call where the end-to-end encrypted password is transmitted. Strictly speaking, this are two keys, because Gemini is authenticated by another key. This is also MAC Hash called.

Calling: Instant Perfect Forward Secrecy

You can renew the encryption anytime just with pressing the MELODICA button. That means: the paradigm of "Perfect Forward Secrecy" has been extended by two components. On the one hand, one can define the end-to-end passphrase manually and also renew the password "instant" - at any time. Therefore, it is spoken of "Instant Perfect Forward Secrecy" (IPFS).

Compared with many other tools, those offer only one key per online session, or you cannot edit the encryption phrase manually.

Symmetric Calling: Send a Call though an existing Call

As a further feature in Goldbug you have the opportunity to send a new Gemini through the channel of an existing Gemini. Here, the end-to-end key is sent by an end-to-end connection. The symmetric encryption phrase is therefore not encrypted with an asymmetric encryption (RSA or ElGamal or NTRU, for example) and then through a secure channel (SSL) of point sent to-point, but is itself (symmetrically) encrypted with the existing Gemini and then only sent by the method described.

Finally, in the context menu (right mouse button, go to a friend in the friends list) is a third method for a so-called "Call" added to the MELEDICA Function: 2-way calling. Here is an AES-256 sent by you as a end-to-end encryption to your friend and your friend sends as well as a response an AES-256 to you. Now the first half of your friend and the second half of your own AES is taken and assembled to form a joint AES-256. This is called the method of 2-way security.

This ensures that no third party - if that party would be able to manage to compromise the machine of your friend, a Gemini (or an old Gemini) on his behalf could be sent by a third, external machine (which is actually impossible, since it would mean an unnoticed takeover of a machine or breaking the existing TLS and RSA (or NTRU - or ElGamal) encryption).

By the ping-pong handshake both parties will ensure that both participants are taking their part respectively to each other to agree on a secure end-to-end password - and generate it “Fifty-Fifty” - in the two way calling process.

The secure transport encryption occurs when a sender generates a (manually) defined symmetric key (message) - encoded with an existing symmetric key (layer 1) - and then additionally encrypts it with an asymmetric key (layer 2). And this packet is sent through a secure SSL/TLS-connection (layer 3). Three layers of encryption ensure, that your message is kept safe.

The options for the end-top-end encryption passphrase • firstly to edit it manually • second, to renew it every second within a new call, • thirdly, to send the password through an existing end-to-end encryption, and • fourth, and finally, to be able to generate the end-to-end password in a two-way process, makes it attackers thus very difficult to break the end-to-end encryption of the Goldbug MELEODICA function.

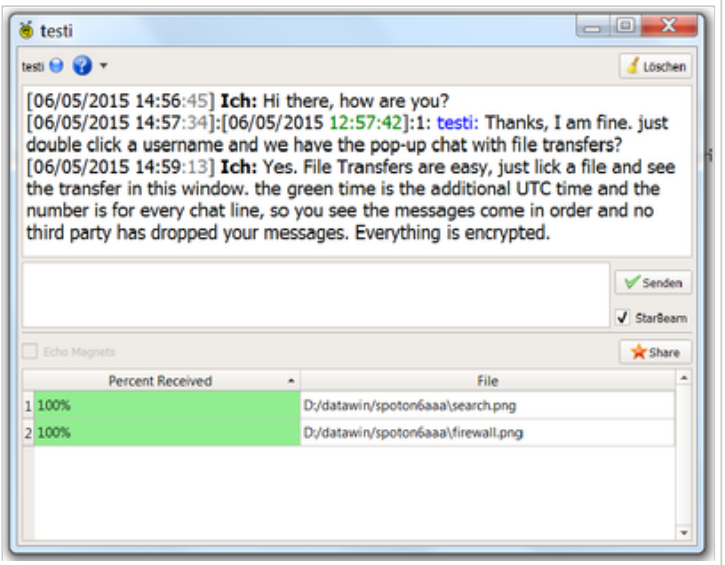


Figure 11: GoldBug Messenger Chat Pop-Up Window

Two-Way-Calling: Define each a

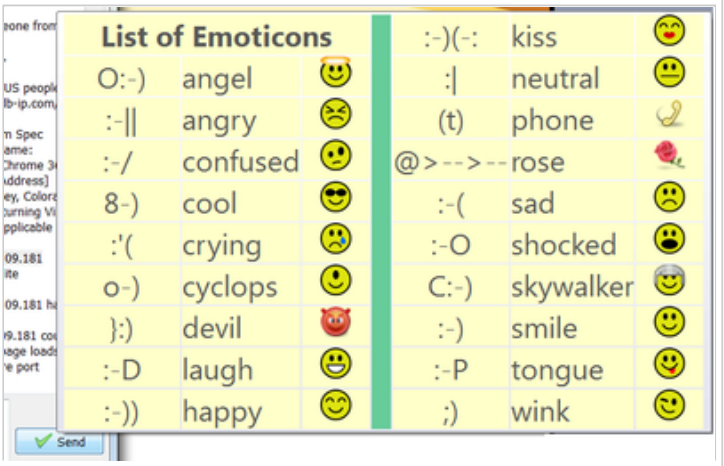


Figure 12: List of Emoticons in GoldBug Messenger

Half of the end-to-end encryption key

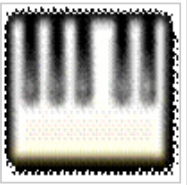


Figure 13: MELODICA: Multi-Encrypted Long Distance Calling

From "Perfect Forward Secrecy" (PFS) not only "Instant Perfect Forward Secrey" (IPFS) has derived, but become a "2-Way Instant Perfect Forward Secrecy": 2WIPFS.

Thus, the Goldbug MELODICA function has PFS and the important element of end-to-end encryption decisively developed with this process implementation: The encryption is not new, but merely the method is implemented in a sophisticated process to provide security.

End-to-end encryption is a simple pressing of a button as easy as calling via phone: Simply pick up the phone or hang up. At any time, the communication is encrypted asymmetric and symmetric - end-to-end encryption can be easily switched out - and be replaced (within a SSL channel) by asymmetric or symmetric encryption. This is a new architectural standard that established this method of calling with MELODICA.

The protocol for the plain two-pass key-distribution system is defined as follows:

1. A peer generates 128-bit AES and 256-bit SHA-512 keys via the system's cryptographic random number generator.

2. Using the destination's public key, the peer encapsulates the two keys via the hybrid cryptographic system.

3. The destination peer receives the data, records it, and generates separate keys as in step 1.

4. The destination peer transmits the encapsulated keys to the originating peer as in step 2. Once the protocol is executed, the two peers shall possess identical authentication and encryption keys. Please note that duplicate half-keys are allowed.

FS-Calling: Calling with Forward Secrecy

Since version 2.7 GoldBug Messenger supports Perfect Forward Secrecy as well for Email. Chat now has as well Perfect Forward Secrecy (FS) now extended with asymmetric keys (as end to end encryption for chat with symmetric keys was already given = "Calling", see above). While the chat is with the permanent chat key always encrypted, we saw above, that a symmetric key can secure this chat with a new layer of end to end encryption. This symmetric key - a kind of AES password - was sent through the permenent asymmetric chat keys. Now the calling feature - to secure the chat with end to end encryption - has been extended: Forward Secrecy is also implemented for calling within the chat over symmetric session keys. This means, you send through your permenent asymmetric chat key to your friend a pair of a session based asymmetric chat keys and use these then to send a symmetric key for the call. In the end you use the symmetric key (for this call again), but the transfer of the password online is created over a) the permanent chat key b) then over the sessionbased chat keys (forward secrecy). Simply send a asymmetric session key (Forward secrecy) through your asymmetric permenent chat key to send in the end a symmetric key (e.g. AES) through the session based forward secrecy key.

Additional Security Feature: Socialist Millionaire Protocol

While Goldbug encrypts the messages three times, first, the message is sent in a secure TLS/SSL Channel, second, every message is asymmetrically encrypted (e.g. by RSA, NTRU or ElGamal), and third, you have the option to “call” with the “MELODICA” function to set an end-to-end symmetric encryption passphrase (with several methods like call within a symmetric encryption or to choose the two-way calling) – it has fourth additionally a further method for security implemented, which is called “SMP” – Socialist Millionaire Protocol (see Wikipedia for further descriptions).It is an asynchronous implementation of the Socialist Millionaire Protocol as defined by https://otr.cypherpunks.ca/Protocol-v3-4.0.0.html.

For the SMP-Process you open up a personal pop-up chat window and find the question mark icon next to the username on top. Define a Password with the selection. Then ask your chat friend to set the same password. Third, you click the “Verify” selection. When both participants have set the same password – respective have the same hash of the same password – then the icon of the question mark changes to a “lock” icon.

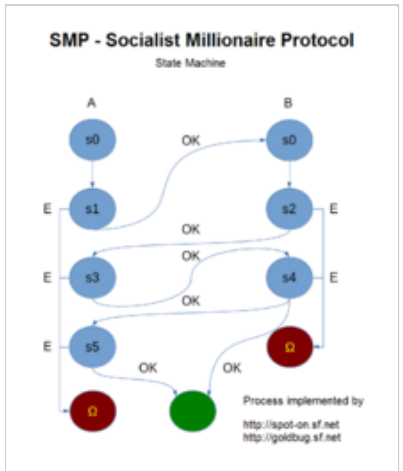
The idea behind it is to ask in the chat a question to your friend like “What is the name of the city we traveled last year?” or “What was the name of the restaurant we met the first time?” etc.

Both participants normally sing the messages with an RSA (or else) Algorithm to verify, that the used encryption key is from the original sender. But in case the machine would be hacked or in case the Encryption Algorithm would be broken, with the Socialist Millionaire Protocol (SMP) – Process you can authenticate a friend by just entering at both sides the same password.

Please be sure, no to send the password over the chat, but to describe a situation, which leads to the same password. For the first time to test the function both participants could use the word “test” to enter it in the SMP process.

SMP is just another option to authenticate your partner with a common secret.

GoldBug does not distribute zero-knowledge proofs during the various data exchanges. Also, GoldBug uses the SHA-512 of the secrets as the x and y components. Assuming that Alice begins the exchange: Alice: 1. Picks random exponents a2 and a3 2. Sends Bob g2a = g1a2 and g3a = g1a3 Bob: 1. Picks random exponents b2 and b3 2. Computes g2b = g1b2 and g3b = g1b3 3. Computes g2 = g2ab2 and g3 = g3ab3 4. Picks random exponent r 5. Computes Pb = g3r and Qb = g1r g2y 6. Sends Alice g2b, g3b, Pb and Qb Alice: 1. Computes g2 = g2ba2 and g3 = g3ba3 2. Picks random exponent s 3. Computes Pa = g3s and Qa = g1s g2x 4. Computes Ra = (Qa / Qb) a3 5. Sends Bob Pa, Qa and Ra Bob: 1. Computes Rb = (Qa / Qb) b3 2. Computes Rab = Rab3 3. Checks whether Rab == (Pa / Pb) 4. Sends Alice Rb Alice: 1. Computes Rab = Rba3 2. Checks whether Rab == (Pa / Pb) If everything is done correctly, then Rab should hold the value of (Pa / Pb) times (g2a3b3)(x - y), which means that the test at the end of the protocol will only succeed if x == y. Further, since g2a3b3 is a random number not known to any party, if x is not equal to y, no other information is revealed.



State Machine Process for the Socialist Millionaire Protocol (SMP) implemented by GoldBug.sf.net Instant Messenger (http://goldbug.sf.net) and Spot-On Applikation (http://spot-on.sf.net).

Anothertastic

TBD.

P2P E-Mail: without data retention

In addition to the chat and group chat function of the Goldbug Messenger there is an integrated e-mail system and this extends the communicational functions to a communication suite.

The e-mail client is peer-to-peer based, i.e. the e-mails are sent over the network encrypted connections.

Further, the email client is also able to handle regular email with POP3 and IMAP.

The p2p email network is provided by the integrated architecture of the spot-on kernel. As shown, the e-mail function uses a different encryption key as the chat feature.

So you can to chat to a friend, but refuse to e-mail with him by not giving your encryption keys for email. It makes sense, however, to always copy all the keys as a whole (“Overall-key”), then you have your friend in all the functions present (in addition: also the URL-key and the Rosetta-key will be exchanged, two functions that will be described later).

Of course, the security of a Repleo can also be used for the e-mail function, if you do not want to expose publicly his e-mail key.

To this end, there are two different methods for p2p email:

One method is that a third, common friend is used to store the emails there in his cache. Basically, the emails do not require a central server, it can be at home, just a third friend who remains continuously online. It therefore makes sense to have more than one friend in your own friends-list and to network with other common friend friends, who can act as a buffer. Since all the e-mails are encrypted, the friends who make a cache function cannot read your emails. You have the choice of whether the e-mails are authenticated or not authenticated, that means you send the emails just encrypted - without proof, that the encryption-key belongs also to you. This proof is done with a second encryption key for authentication and signing the first encryption key.

The interesting thing about the Goldbug e-mail function - and here it might differ from other p2p Email implementations - is that it is possible to send email also to friends, who are offline. And, that is it hybrid with the POP3/IMAP Email system, so it is currently a model to replace other regular e-mail clients, when the function - respective the GUI - has been more elaborated to the current standards by a subsequent Qt-E-Mail-Client Team.

In summary, GoldBug provides two e-mail models for distributed e-mail. Endpoints may optionally define themselves as institutions or post offices, or both. A brief description of e-mail institutions follows. E-mail institutions are artificially characterized by addresses and names. The information is not considered secret and several endpoints may identify themselves identically. It is the responsibility of an institution to accept subscribers, that is, public-key pairs. Please note that a separate model could consider the use of signature keys instead of key pairs. The data that an institution houses is stored in encrypted containers. Unlike physical institutions, GoldBug institutions are only allowed to read the signature portions of e-mail letters. The signatures allow verification of deposits and withdrawals. The sole difference between e-mail institutions and e-mail post offices is that post offices require the distribution of public-key pairs.

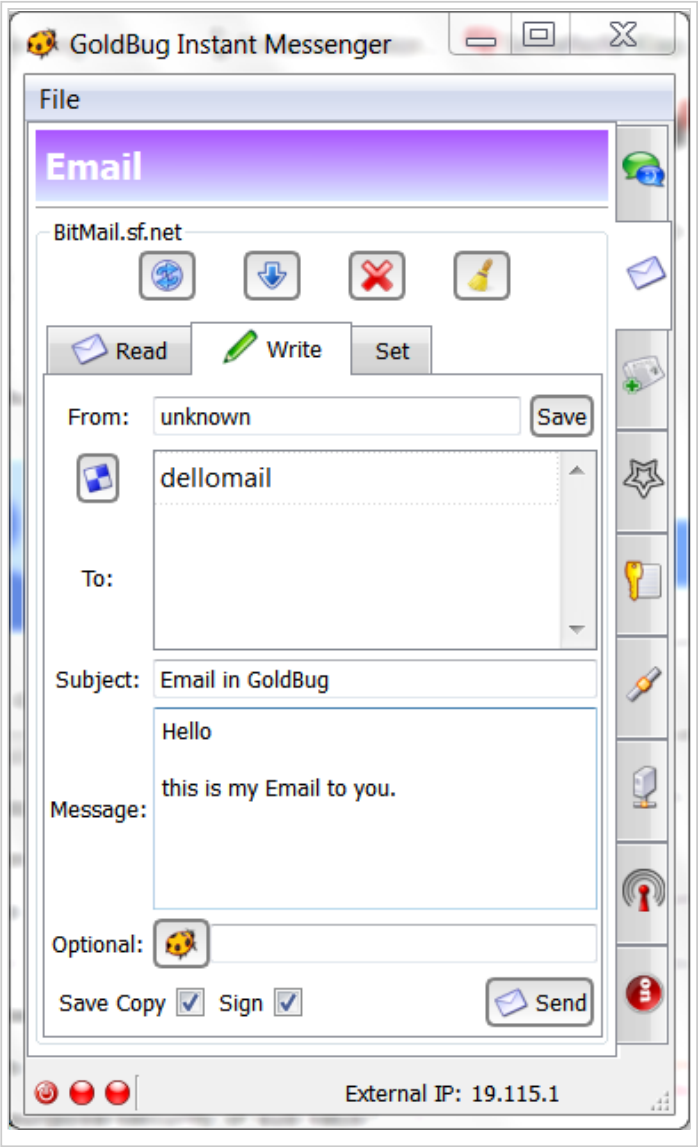
Feature: Set additional encryption with a "Goldbug":

Not only the software is called Goldbug, but also the function of the integrated e-mail client to set an additional password on the e-mail.

E-mails to which a "goldbug" password has been set (see later below the description of the file transfer function "StarBeam", here's the additional password called "Nova") can be read by the recipient only, if the corresponding Goldbug-Password is known - the “golden key” to open the e-mail. Thus, you should inform your friends when you send them e-mails that still need an additional password for opening.

An example may be found with the e-mails to your wife: Always encrypt e-mails to your wife additionally with the set Goldbug-Password, which is the city name, which hosted your wedding with your wife.

Figure 14: E-mail tab



To activate the care-of (C/O) caching function, check box "care-of" must be activated in the sub-tab "Email Settings". Then a third friend of two other friends will allow to cache the e-mails in the own client, when both friends are participants in the e-mail own e-mail-contact list.

The second method to cache emails in the p2p email network is to create a virtual email institution. For this purpose, it is also necessary to activate the C/O function with the check box, described above.

Next step is to create a virtual e-mail institution and to announce the created Magnet-URI for E-Mail-Institutions to the friends, which add the Magnet of the E-Mail-Institution to their own client. The last step is, that the public key of the email friends must be added to the node, which provides the E-Mail institution.

The advantage over the first method, however, is that the public email key of the node, that establishes the institution, must be NOT given or known to anyone.

E-mail attachments can also be attached in any case and are automatically encrypted.

C/O and E-mail Setup institutions

The following example describes, how such a C/O function for e-mails and the creation of a virtual email institution is implemented step by step.

1. First, activate the C/O function in the tab for E-Mail-Settings.

2. Create an institution and choose a name and an address for the institution.

3. Example: Name = "GB mailbox" and address = "Dotcom"

4. Add the E-Mail-Key of your friends in your client and let your friends add the E-Mail-Magnet-URI of your institution in their clients. The magnet will look similar to this:

magnet:? in = GB mailbox & ct = aes256 & pa = Dotcom & ht = xt = urn & sha512: institution

You realize an E-Mail-Magnet at its ending: “URN = institution”. Then you know that the magnet is not a buzz-group-chat magnet and not a star-beam-magnet for file sharing - because they have the extension "buzz" or "starbeam".

After sharing the magnet-URI for Institutions and adding friends E-Mail-Key, your node will cache the emails of your friends - even if necessary for recipients of an E-Mail that should be offline.

You (as the creator of an E-Mail-Institution) need not to expose your own e-mail key with your friends / subscribers of your institution.

You can exchange the Magnet-URI of an E-Mail-Institution in a group chat room (based as well on a Magnet URI with symmetric encryption details). The exchange process for E-Mail-Key & E-Mail-Magnet must therefore expose no further identities.

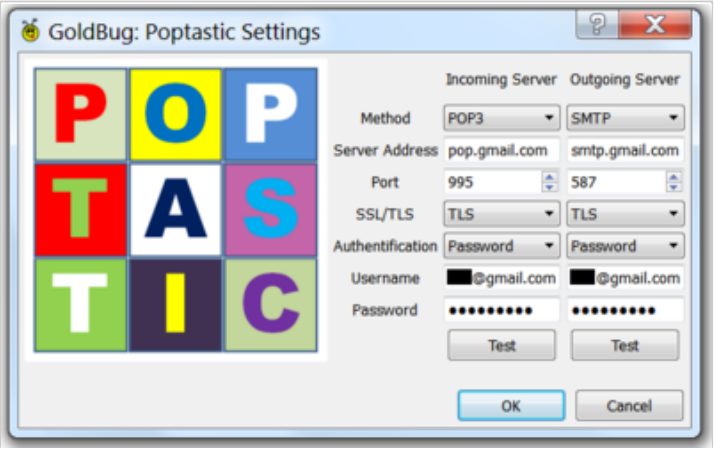
GoldBug E-Mail client - Encrypted Chat and e-mail via POP3 and IMAP

In addition to the encrypted option exists - if the POP3 or IMAP settings have been defined in the Poptastic settings - GOLDBUG has the chance to send plain text emails and receive. This GOLDBUG is an e-mail client like everyone else. Unencrypted messages can thus be received well and who wants to send a message unencrypted POP3 or IMAP, is beside the mail form in which an e-mail can be written also a check box can be sent with the plaintext.

Poptastic Feature

Poptastic is a global innovation in communications - Encrypted Chat via POP3 in the chat and e-mail program GOLDBUG. The encrypted chat - and of course encrypted e-mail - POP3 (or IMAP) can be described as follows: With the Poptastic function can now all e-mail accounts, for example, from Gmail, Yahoo or Outlook.com -mail with GOLDBUG! are encrypted - end-to-end asymmetric - and hybrid complementary symmetrical. The trick: Each POP3 or IMAP server can also be used for encrypted chat. So why not use a dedicated chat server or secure chat logs with plugins for encryption, when you can just use his e-mail address for the chat and also e-mail? The 30-year-old POP3 protocol and thousands of email servers can now be used for encrypted chat with this app. The e-mail server is simply converted as a chat server. To this end the chat Night layer is converted into an encrypted e-mail, sent via POP3 or IMAP, and the receiver is converted back into a chat message. Since the GOLDBUG Messenger is also an e-mail client, the encrypted message exchange also works via e-mail. The program automatically detects whether it is an e-mail via POP3 or a chat message. The chat and email Poptastic are proxy-capable and can therefore be operated even from work, university or behind a firewall, and over the network gateway. If you logged in a browser in his e-mail account, you can see how the encrypted message looks like.

Figure 24: Poptastic Settings: Encrypted Chat via POP3 and IMAP servers



Poptatsic Seetings

It remains to be seen how users and POP3 Aministratoren use this new function mode. Finally, you can promote encrypted e-mails and chat only retrieve encrypted is to be welcomed also from a POP administrator. If contacting the age of e-mail servers to be increased, this does not turn out the possibility of encryption, such that the contact time of a POP3 server should rather be lowered to chat lines per second, rather than every 3 or 5 to permit seconds.

Otherwise, the function of a fully encrypted e-mail client using POP3 or IMAP is expandable or integrated into other e-mail clients through the use of an echo kernel. In a future development @ -mail will be sent to this e-mail and chat clients via dedicated echo chat server. Thus, the previously used by GOLDBUG echo protocol POP3 server would just can thus be replaced by other chat servers without encryption - at least for a use of a / This e-mail & IM client out. Due to the encryption, it is thus also (with increasing encryption) a (currently more unrealistic) Farewell to the web mail - if not so even a further suggestion and idea implementation for the increasing encryption of private user communication has been set.

The complementary symmetric end-to-end encryption via POP3 can be used as the Echo protocol not only as perfect forward secrecy, but can also "instant" be renewed every second. Therefore, it is also spoken here (as above) Instant Perfect Secrecy (IPFS), which is now possible via POP3 and IMAP! Finally, there is also Poptastic an option, the password via the channel of an * existing * End-to-end password to send - and thirdly, that both parties define one half of the end-to-end password and authenticate to each other and made even safer.

For users to chat certainly an exciting, new way encrypted via this protocol.

Detailed description of the setup options:

From the main menu "View / View" Messengers of GOLDBUG your own e-mail address and the POP 3 or IMAP server details are stored. These are the same data that are entered eg Thunderbird e-mail client or Outlook, for example:

```
Incomming Server Server: pop.gmail.com
Port: 995
TLS
Username: mygmailname@gmail.com
Password: ********

Outgoing Server Server: smtp.gmail.com
Port: 587
TLS
Username: mygmailname@gmail.com
Password: ********
```

Please press each test button to check the functionality of the server entries. Then save the entries with the "OK" button. (If it is used in the selection menu instead of POP3 or IMAP, the value "Disabled", the program does not send encrypted emails anymore). Then you can all Jumpsuit encryption key (or key for the Poptastic) exchange with a friend for a new installation. If you and your friend have changed and entered the jumpsuit key or the Poptastic key, the chat can start in a running kernel by default. In Gmail, you should, if necessary, set the option in the Web that retrieved POP3 messages are deleted from the INBOX. To connect, you should also define the security setting in Gmail so that all local e-mail client can connect to Gmail: (1) Settings / Forward and POP & IMAP / POP Download: Enable POP for all mail (2) Settings / Accounts & Import / Change Account Settings: Other Settings / [New window] / Security / Access for less secure / unkown Apps: Enabled. It is recommended that, if necessary, set up an extra email account for a first test and further use. Note that new e-mail accounts are possibly limited to 30 days for the sending of e-mails (eg Gmail for 500 chat lines or emails per day).

E-Mail-Forward-Secrecy Feature

Email has been extended since version 2015-09-26 of the kernel (GoldBug Version 2.7) with Perfect Forward Secrecy, short: FS. GoldBug and the underlaying kernel architecture is worldwide the first E-Mail client, which supports Forward Secrecy. You send to your e-mail partner over the symmetric encryption of your permanent email key a sessionbased (forward secrecy) symmetric key. When your e-mail partner confirms the request and sends his session keys back, then both email participants can use sessionbased asymmetric keys to secure the email communication. Forward Secrecy has been also implemented in chat for the calling feature (see above: calling with FS). When you write now an email, you can select 4 modi of encrypting it.

- Normal: The email is sent as is within the system (echo or poptastic), that means the regular permanent symmetric email key is used, to encrypt your message.
- Forward Secrecy: Over the encrypted connection sessionbased forward secrecy keys are used - that means you send your message encrypted with sessionbased keys within the permanent email key. This adds another asymmetric layer to your email encryption
- Pure Forward Secrecy (Pure FS): The message is sent and encrypted only over your sessionbased (symmetric) email keys. This can be called the option to create within the peer-to-peer email "instant" email-addresses and post boxes, which can be deleted after your session. One-Time-Email-Accounts thanks Forward Secrecy.
- Goldbug: Goldbug sets a password on the email (e.g. AES, symmetric encryption) and you need to inform your email receivers about the password in an oral way. This message is as well sent over your asymmetric email keys.

In case you click the Email-text checkbox-option "plain", all encryption is set back to plaintext - so that the receiver and all intermediate transmitters can read it anytinme.

Echo-ed IRC

The Goldbug Messenger has besides to E-mail and Chat also as already mentioned a group-chat feature. This works similar to an IRC chat. The transmission of messages to all group participants will be here again fully encrypted using the echo protocol. Ultimately, all the participants will be able to read the content in a group-chat, who know a particular symmetric end-to-end key, that defines the chat room in the p2p network.

Therefore, it is spoken of an echo-ed IRC (or short: “e*IRC”) – which opens to IRC chat new options, because the transport route of the e*IRC chats are also always encrypted - as today regular POP3 or IMAP e-mails have also at least an encryption for the transport, e.g. with TLS 1.3. Hopefully the traditional IRC-chat will therefore increasingly take account of such safety features. The e*IRC-chat can represent a model of a new generation of IRC.

The encryption details of the group-chats are again defined by a magnet-URI (defined ending: URN=buzz).

At the start of the program Goldbug the developer-chat-room is opened, which can serve as an example for echoed group IRC-chat.

To join a private channel, just type in the room name or use the above-mentioned method of magnet-URI links. The magnet link next to the room name has additional values for the encryption embedded such as keys, hash or cipher for encryption type. If you just typing the room name, and add no magnet-URI, the additional encryption details are set to the value of “0000” and the encryption of the room is only based on the room name.

When you have entered all the values, press the "Join" button – or: if you have inserted a magnet-URI, then use the pull-down menu and select "de-magnetize". The magnet is again broken down into its individual components and encryption details and the chat room is created and entered on the basis of the given encryption values.

If the room is open, you can save the chat-room as a bookmark or at any time printout the corresponding magnet-URI of your chat-room. Also you can send Magnet-URI-bookmarks to your friends to invite them into a room.

To send a message, write some text and hit the send button.

The e*IRC chat room can be public or private, that depends on to how many people you are sending the magnet-URI or the individual encryption values. To announce a public e*IRC chat-room you can add a Magnet-URI on your website and everyone knows, how he can come in your chat room - with "de-magnetize"!

Ultimately, it works like having a chat - with the only difference that the ISP and more rooting server cannot look into the communication because it's encrypted – comparable to your connection for online banking.

So with the echo protocol it makes no difference whether you are talking to friends or your bank manager.

If you want to use the chat room as a private room, you can even share with friends the magnet-URI for the chat-room - without exchanging each other’s public (asymmetric) key for chat. Just create a one-time-magnet and -room and protect your public chat key!

This feature is one of the peculiarities of the Goldbug program that you can chat easily encrypted without having previously to exchange asymmetric keys or you can can swap asymmetric keys in a private IRC-room – as a protected asymmetric key in a private chat-room (based on symmetric keys (Magnet-URI keys)).

Goldbug allows a secure key transfer with the Repleo and additionally the key exchange over a one-time magnet (OTM) for a private chat room - your public key does not need to be public!

While other applications share the public-key with all friends or even in a DHT and partly also relate its own IP-address to the keys - that above presented architecture for the transport of encryption keys is much safer and forward-looking.

Figure 15: IRC group-chat within the echo

FileSharing: with Starbeam

As in any messenger file-sharing between several persons - or a file transfer between two defined groups of people – is provided in Goldbug. The file sharing function is called “Starbeam”.

For this purpose, it is necessary to point to the following steps: • Adding or Creating a SB-magnet-URI • Optional: encrypt the file with a pass phrase called "Nova" • Optional: encrypt the file with the file encryption tool. • Select the file and a SB-Magnet: How to transfer the file encrypted.

Figure 16: Starbeam tab for file transfer

The tab "StarBeam" for the file sharing consists of three sub-tabs: one for uploading, one for downloading and one for creating or adding a SB-magnet. Many users still know it by an Emule or Torrent Client: more easily it cannot be: upload, download, and a tab for pasting the magnet-URI.

SB-Magnets and Novas

A magnet-URI is a standard that is known from many file-sharing programs (often in the Gnutella network) and also eDonkey / eMule ed2k-links or torrent links corresponds. The evolution of the magnetic URI standards by the GOLDBUG Messenger underlying spot-on library lies in the design of the magnet URI with encryption values. Magnets are so used to create a bundle of cryptologic information or keep together. SB-magnet URIs are therefore referred to the community as a crypto-Torrents, since they can be linked to a web page as a torrent link and access to a file - can be linked - or even as a channel for different files.

Through this dual-use effect a magnet cannot be assigned to a single file or a specific IP address. A file name does not appear in the crypto Torrent or SB-magnet, as yet, even at the - is more advanced example of Offsystem.sf.net links or Retroshare.sf.net - compared with Gnutella, eMule and torrent link. However, while numerous opinions see the link of Gnutella, eDonkey and Torrent Links-critical, consists in a collection of encryption values no reason to discredit these values.

Your homepage or independent portals Find Starbeam so advanced technology. In addition to the strategic decisions of the selection of a link standards but it comes at the use aspect to maintain the security of the file transfer between two private users.

For the flow of private file transfer from friend to friend some more information: Before you will send a file, you can consider if you simply appending send an email to an email within GOLDBUG. This is the version of choice when the file is smaller than 10 MB. Larger files should be operated only on the Star Beam function.

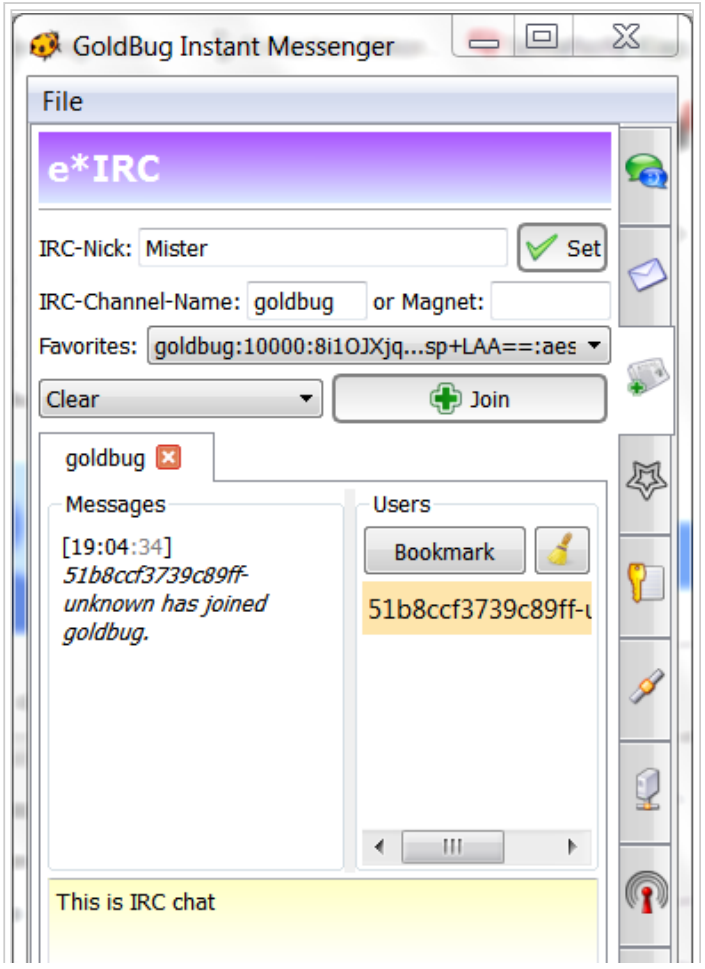
Before shipping You can also consider to encrypt the file on the hard disk. These holds the GOLDBUG Messenger in the main menu under Tools / Tools tool for file encryption ready. With a double passphrase, the file is encrypted in it. Some pack the files into a zip and encrypt it before sending or uploading. The zip encryption is very easy to crack 96 bits, so far so you should use a key as it is now recommended for RSA with 2048 bits. No matter how you put your file now vorbereites - such as it is, as plain binary, or encrypted with the GOLDBUG tool from Starbeam - yes it is encrypted again several times with the echo protocol.

Just like you can put an additional password to an email at email ("gold bug" mentioned in the e-mail function, see above), you can also click on the file - or more precisely set another password on the used magnetic UIR file transfer. This is called "Nova". Even if the file transfer is successful successful or a third stranger could poop the previous multiple encryption (which is not likely), is introduced with the Nova password an end-to-end encryption, which is safe as long as the common Password is exclusively for both partners under wraps. To send a file using an encrypted channel must be created. This works again (indicated at the end URN = SB-Star Beam) with the creation of a magnet. To file for package file packet is - also file chunk or file link called - transmitted over this channel using the HTTPS protocol (which can based on TCP, UDP, and SCTP also connections). Therefore, it is an interesting question whether a transfer of a large, encrypted file using Starbeam over SCTP, TCP or UDP protocol, ceteris paribus, is transmitted correctly and fastest.

Thus it is clear that in Starbeam no specific file is changed, but are generally exchanged only encrypted channels. It's like a "wormhole" to "Stars" to stay with the term. And this channel is defined by a magnetic-URI link.

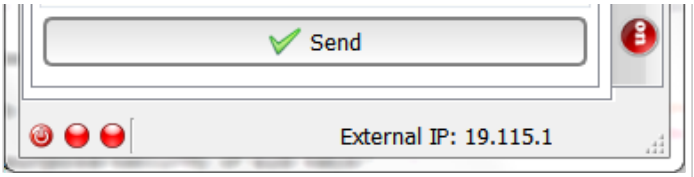
Figure 17: SB-Magent URIs & Novas

Ideally, you have your own magnet URI for each file. That would be then a one-time Magnet (OTM), a magnet is used only once for a file. (OTM thus corresponds to the idea of an OTP - a one-time pad . A string that is only used once OTP is often considered in cryptographic processes as crucial to establish security). You can also use permanent magnet but a URI, then it's like, a subscription video channel in which, for example, every Monday a file is sent.



This opens eg also Torrent portals new possibilities, it must no longer exist portal, linked in the thousands of links. The portal itself requires only a single magnetic UIR in this decentralized network echo, in order to send Consecutive then gradually a file after the other through the wormhole. Who's afraid that the neighbor connected could disapprove a file transfer, then you need to switch only on p2p and f2f with Echo accounts a Web-of-Trust to create. Connect your node only to a trusted friend by finding all the credentials of the echo accounts for sharing and a magnetic-UIR file for your channel. At once you have transferred a file from the magnet URI, so you can delete or keep the magnet URI. You Erstellst the magnet as OTM and activate the check box for OTM, then it deletes itself after file transfer by itself. Man, that's like Mission Impossible.

So you can share with your sister and securely transmitted over the Internet without having to unencrypted upload it somewhere your journal your vacation. The tool of GOLDBUG-File Encryptor you can of course also use it if you want somewhere to upload to an online hosting a file. However, as these files if necessary to control and encrypted files are marked with a question mark, although it should be



Echoed IRC = e*IRC

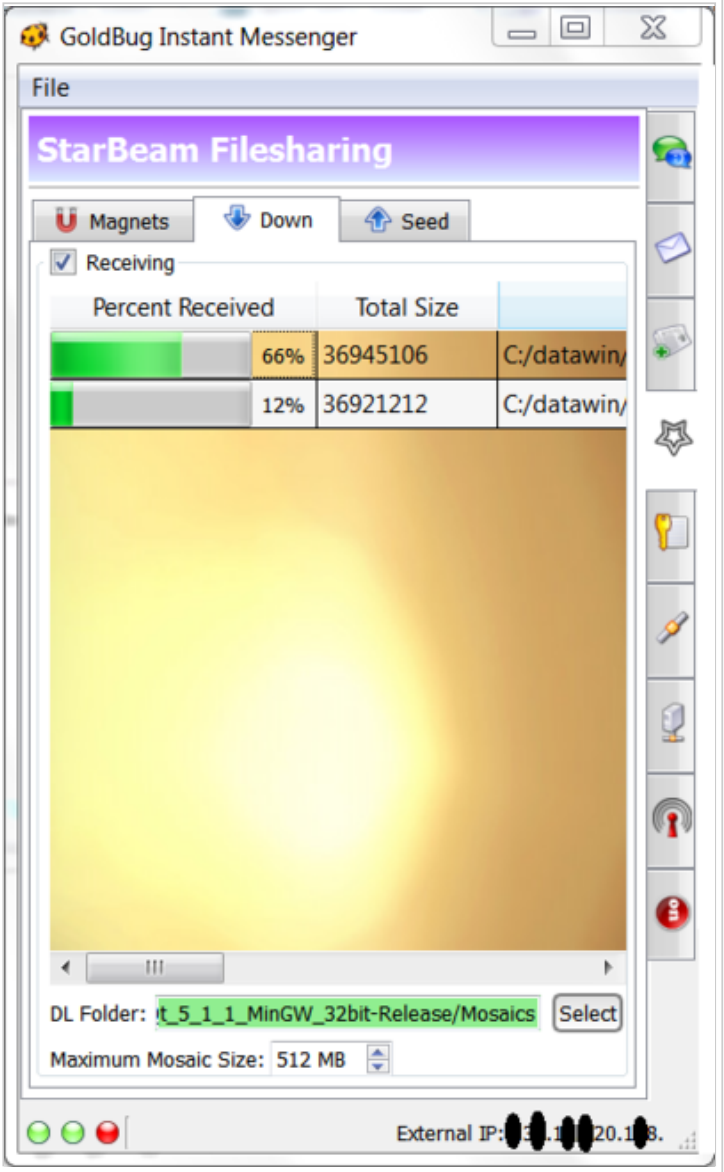


Figure 16: StarBeam-Tab

an exclamation point, it makes sense, the encrypted file right from point to point, from friend to friend to transfer over GOLDBUG.

As mentioned, it is recommended that called on the file transfer at least one. Nova added as additional passphrase. For if the transmission of the SB-magnet URI should be monitored - You must crypto Torrent somehow transferred online to your friend - then everyone who knows the magnet URI can also receive the file as well.Therefore, it makes sense to protect the file with a Nova - a password that have changed both friends possibly orally, in the past or through a second channel. The Nova also builds on the end-to-end encryption standard AES on (if you do not think up you own a passphrase). And it must - before - the file transfer begins, have been stored in the node of the receiver.

If a recipient has a file packet, a chunk or link received, he is able to upload this again - in other magnet URI channels - or to give it again in the same channel. This is similar to a rewind function: The file is simply played back again again like on a cassette recorder or MP3 player via the echo network. The file can be also sent many hours or days later. Anyone who has obtained a copy of a magnet URI channel becomes a satellite television, and can the data into a wormhole or better: import Starbeam Magnet URI again. To perform the transfer, you need only one connection to a neighbor or friend and can secure them with an echo-account, so that only friends can connect with each other.

The transmission of the Echo protocol is more effective than using a protocol similar to the " Turtle hopping "(see Wikipedia) to run because, depending on the configuration of the echo-network here (Full echo, half echo, Adaptive Echo, Super Echo) and the basic encryption nodes with only low bandwidth do not necessarily act as a bottleneck, but on further optimize echo paths the desired download speed.

Upload and Transfer a file

If you have a magnet URI defined or generated, it appears not only in the sub-tab for the magnets, but also in the table in the sub-tab for upload / seed. Select the check box of a self-magnetic. Also choose the file. Finally, you might even decide if you transfer to an additional password - want to place - called Nova. For the first test, you can skip this first time. The chunk size (Pulse-size) can be left as pre-defined. The echo is transmitted as an HTTP post or -Get, corresponds to the transfer of a website. If the pulse size is made larger, the website is as it were longer transmitted. Then push the button "Transmit" / "Transfer". Finally, the magnet URI copy and send it to your friend. If he has copied him, you can start the transfer with the deactivation of the pause function. The magnet URI can printout in the right side splitter to transfer table.
Figure 18: Transfer a file

Download a StarBeam File

To load a file with Starbeam, you need turn a SB-magnet URI or sometimes colloquially referred to as Crypto-Torrent. This you can find on websites linked or can you this from a friend who wants to send you a file obtained. Copy then the magnet URI in the sub-tab for the magnet URIs easy. Share your boyfriend that you have inserted the magnet URI and he can start the transfer. Previously, you should not select the check box "Receiving" / "reception" in the download sub-tab. Then should start

the download, once a sender sends a file via the echo and through the channel of the magnet. With the other settings on this page you can also define the size and the path to the download area. The successfully downloaded parts are called Mosaics. The files to be transferred are links (or in the community also: Chunks) called.

Figure 19: Download files

If a file does not even have been successfully transferred, this can be checked with the Star Beam Analyzer tool. This determines whether all mosaics are available or whether left or chunks missing. If there are missing links, the SB-analyzer creates a magnetic URI, the friend can enter into his upload tab again. Then only the missing links or chunks are sent again. The file would also complete, if the transmitting station ("Resend" =) three times a day for the echo to the "Rewind" - sending function. Note that a magnet is a channel and existing files will be renewed in your mosaic path then when no one-time magnet is used. Create Starbeam magnet URIs so new ways of thinking when it comes to the use of crypto Torrents about the echo protocol?

Create an initial setup to a neighbor

Communication Methods

GoldBug supports SCTP, TCP, and UDP communication methods. For TCP-based communications, OpenSSL is supported. GoldBug distributes data with or without SSL/TLS. Please note that magnet distribution violates this principle and therefore requires SSL/TLS. Communications between the GoldBug Kernel and the GoldBug User Interface also require SSL/TLS via TCP. In essence, the application is generally method-neutral.

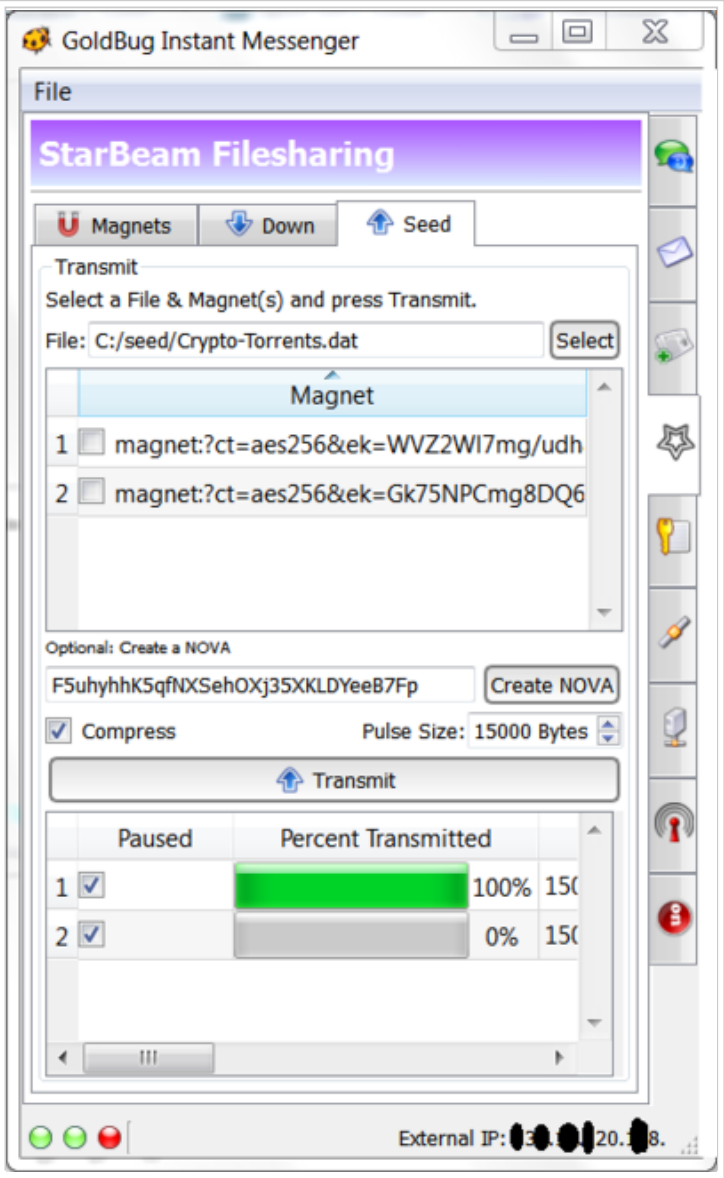


Figure 17: Magnet and Novas for StarBeam File Sharing

Adding a neighbor

As a very first profile is set up, it has already been explained above. Enter the nick name twice and a 16-digit password. Done. Optionally, select a question / answer phrase instead of the password. In the following, it is now a setup of the network. If you explore the GOLDBUG Messenger for the first time, you will be connected through the Project Server. Friends of yourself as well, so that the software can be tested by you. Then - if the basic functions are clearly - plan advanced users certainly also the use of a private chat server or the connection without chat servers directly between two friends.

Therefore, the next steps explain the

• Connection to a neighbor / chat server,

• create your own chat server and listener

• and other details that can be displayed in the non-minimum-view.

To make it easier and easier for the beginning to make it, choose the main menu, select "minimal View". Go then on the tab: "Connecting neighbor". This shows an input field for the IP address of the neighbor or the web server where a spot-on kernel runs and a friend also uses a GOLDBUG Messenger.

Figure 20: Adding an IP address as neighbors.

I enter the IP address of the neighbor nodes in the field. With the points three digits of the IP address are separated. Umfast a block with two digits, eg 37100100100, then the 37 be placed anywhere in the first block or be entered as 37 on the first two positions. Then press the "Connect" button.

The IP address is then deposited on the default port 4710.

If an error message appears, then the IP address is already entered. To delete all the neighbors, you can then the button "Delete all neighbors" key and enter the IP address again.

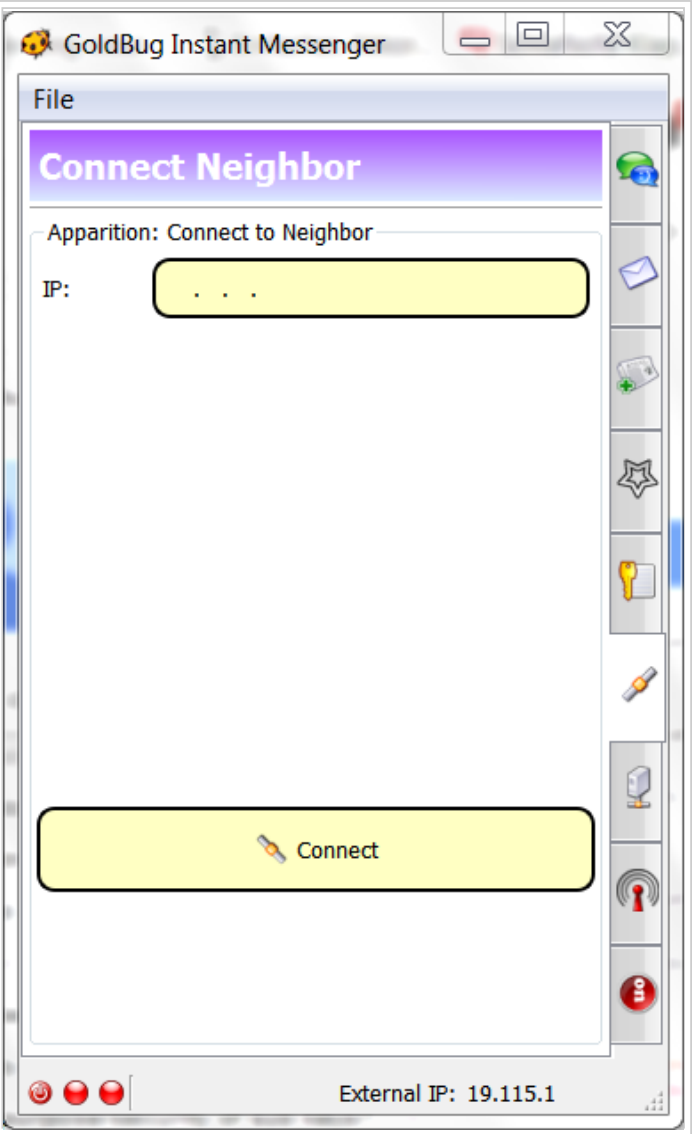


Figure 20: Add neighbor IP (simple view)

Optionally, in the installation path ./spot-on on the hard disk, the file "neighbors.db" are deleted. It is formed immediately new and is then empty. When the kernel is enabled (left, first LED in the status bar is green) and the neighbor is connected (middle LED lights up green) everything is successfully installed and online.

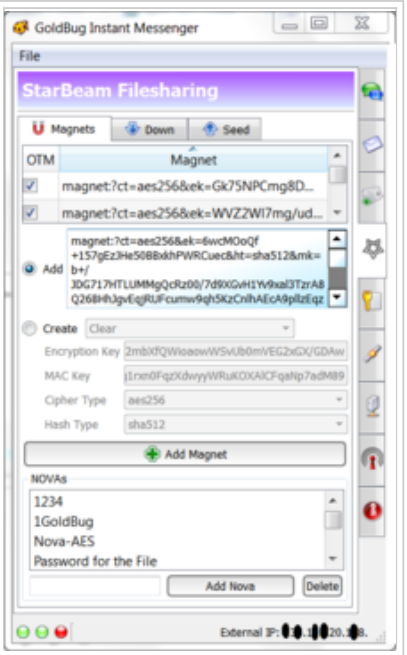


Figure 19: StarBeam-Magnets added to SB-Tab

Enter an IP address and press the connect button, should succeed. Who wants to see more details, the minimal view also switch to the full view. In this view, it is clear that in addition to the IP address and the port of the IP address can be configured individually. By default, the port uses GOLDBUG 4710th

Furthermore, the client can also be operated via IPv6 and control a listener that the Dynamic DNS is linked. This one is then no sequence number in its IP address but a domain name.

In using the box below additional security options can be set. Setting up a chat server, or spot-on kernel means to establish a so-called "listener", the technical term. This is the default for the TCP protocol, but also for GOLDBUG is equipped to set up a listener on UDP and SCTP protocol thirdly. Both latter protocol are ideal for VOIP or streams.

Therefore, it may be defined in the connection options, if your client should connect using TCP, UDP, or SCTP to neighbors or server. The neighbor listener or the server may waive SSL connections, then the transmission is not over HTTPS, but only over HTTP is regulated. A listener can set the security option to create a permanent SSL certificate. This is the existing SSL for Diffie-Hellman key exchange and -Verhandlungsprozess Not renegotiated at each meeting, but an attacker would have to a negotiation process in the past to know to intervene. However, it may be that the server or listener is renewing its SSL certificate times, so it makes sense if necessary, exceptions ("Exceptions") allow, if you want to create a connection easier and this added security layer does not want perfect.

Similarly, one can, in turn, define the key size for the SSL connection and also determine that the compounds below a certain SSL key size will not be set up. Once thus defines what needs to neighboring SSL key size and the other time is defined which key size you expect from a server or neighbors. Finally, there is the option that the client determines whether it connects to the neighbors with a full or half echo. At half echo the message packet is sent one hop to the Direct connection only to the neighbors. Suppose your friend has set up your web server and also sits before and you do not want that your echo packets go to third and his friends, then you can define the Half echo that your packets will be not more widespread after

receipt by the server , So you chat via a direct IP connection. Both participants see the Half echo of your friend and chatting with the IP address. In the solid echo the chat friend does not have to be an administrator of the node, but can connect multiple clients to each other as a central chat server.

If you want to let GOLDBUG as a client through a proxy in the company, behind a firewall or a proxy of the University or on the run Tor anonymizing network, you can insert the proxy details for a neighbor.

As a client you can connect you to the HTTP protocol from any IT environment, if you can surf in this environment with a browser. It is crucial to address a node Punk on the Web with a GOLDBUG node that will not possibly limited by the Port forth through your firewall or proxy. If this is the case, please still your friend, GOLDBUG chat server on port 80 or port 443 instead of 4710 to set up and this possibly be provided with login details for Echo account and make these available to you ,

Figure 21: Full view when adding a neighbor

If you want to run your GOLDBUG chat about the Tor network, this is also very comfortable, so that a Tor exit node will only see the passphrase of GOLDBUG. Here is the chat server to a normal web outside of the Tor network; occasional participant in the gate community are in the process of designing the installation of a GOLDBUG chat server / listener inside the Tor network.

Since the echo protocol is not necessarily a DHT needs, but just a simple HTTP connection to a neighbor that can potentially be mapped through the Tor network, it is a very simple architecture, chat safely through a proxy or a proxy network to operate.

This is also potential for further testing, experiment descriptions and documentation are given if necessary, to bring the synergies of the clients inside and outside the network forward together and explore information technology. If you want to test or use a proxy, for example, in your company or university with the GOLDBUG messenger, then this is not critical, because it is a SSL / TLS or HTTPS connection established - which is hardly different for the proxy administrators as SSL / HTTPS connection to an HTTPS website in banking or logging into your Web e-mail.

Encrypted traffic remains encrypted traffic and ports 443 or 80 can be achieved at any GOLDBUG friend.

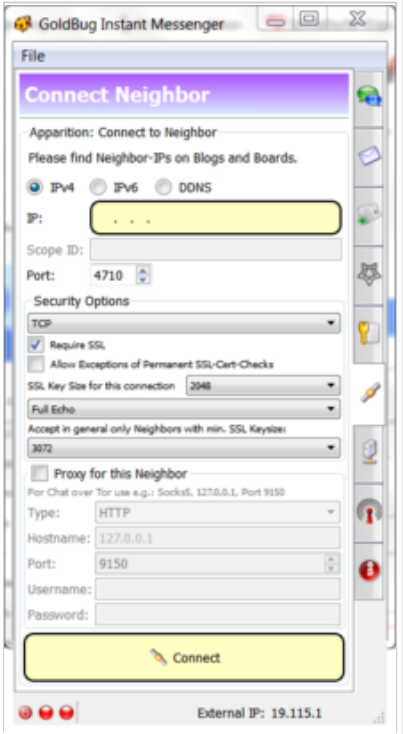


Figure 21: Add a Neighbour IP

Setting-up an own EMPP chat server

Once you are least in the minimal view, a chat server or listener is set up as fast as in the previously described tab a connection is made to a neighbor. Again, for Erinnering: "Connect" in the tab, you connect your GOLDBUG with another node or neighbors, and with the tab "chat server" you create a server or listener, so that others can connect to you. No matter which method you can always send messages when the second or third LED in the status bar and a neighbor is connected. The right (third) LED in the status line thus indicates that you have set up your own chat server on your computer.

Moreover, you will have to enter the local IP address of your machine. This is not the IP address of the router, but the network IP address of the device on which you have installed GOLDBUG. Again, you use the pull down menu selections and can choose the local IP. As a port is then defined automatically 4710. Dücke the button "Set" and the entry of your listener is successful if the third LED lights. If you have a client who is on your server, or you're connected in the "Connect-neighbor" - Tabulator from You to another chat server, or friend, then you can also head "Go Live" button. This is communicated to your chat server via the existing connections show your friends and neighbors and friends as well as their friends. "Go Live" Thus says "Broadcast IP + port" your chat server to your friends and neighbors. Then you can also connect automatically to your chat server. So you have to tell an IP address or you can enter more friends Your IP address manually. Everything is then automatically and your server is to your friends and their friends as a peer available. So Easy A chat server can be created.

Figure 22: Setting up a EMPP chat server - Simple View

The echo protocol from the messaging area or for the chat server creation and referred to as "EMPP" and stands for " Echoed Messaging and Presence Protocol "- certainly based on XMPP protocol elaborated as little regard to encryption applies and due to poor upgrading with encryption capabilities and options even at cryptologists and data protection in terms of the architecture may be true, despite existing Popularity technically antiquated. If you still want to define additional features in

the non-minimal view, is a frequently used function of the echo accounts. Mark in the table to the listener you created, and then give the account credentials a, ie name and password. Share your friends with how the account name and the password is for it and he is when he makes contact with neighboring asked via a pop-up window, enter these credentials. Likewise, you can also back between IPV4 IPV6 and choose when you create a listener / chat server will. Also, multiple chat servers can be created by using a different port is selected. Test several listener on port 4710 or 80 or 443 and decide whether you want to define this listener for friends with an echo account, or for simple connections to be established in peer mode without account log are running. Echo Account define whether to build up a F2F network or a P2P network, because the account credentials you create a web-of-trust, trust only your friends can connect to the login password with. If you are running a peer, you can also click on a LAN party of a closed network knob notify all participants with the go-live that your node has opened a listener for the guests, for example.

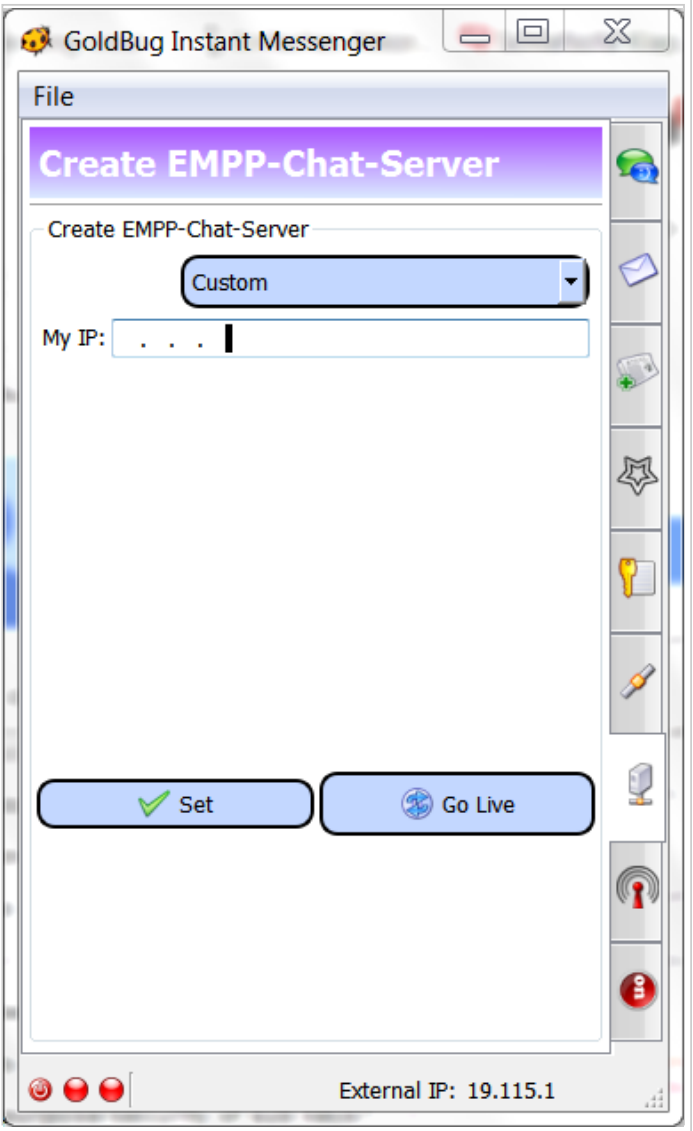


Figure 22: Create Chat Server (Simple view)

Configure your IP address as a chat server: Figure 23

Security options allow in the enlarged view for creating a chat server / listener further define the SSL key size and vorzuhalten also a permanent SLL certificate. Also you can - if you have a permanent, stable IP address - these include in the SSL certificate. These three measures make it attackers from to replace the SSL certificate or fake - because it would immediately recognized when a supposititious other certificate would impersonate the original: for example, because the client does not expect a new, but the old, permanent certificate or because the IP address is missing or is not consistent. The SSL key size defined this.

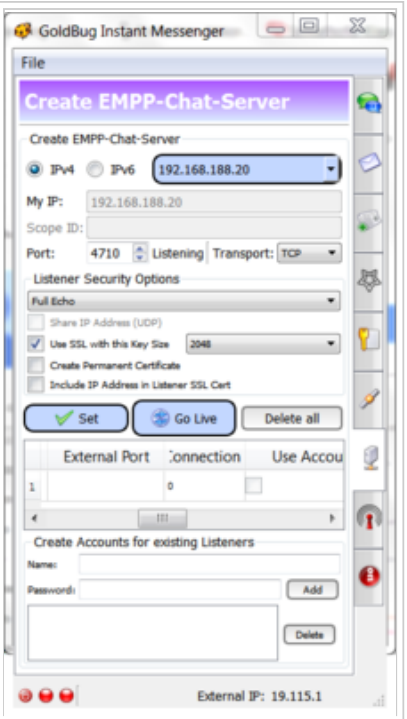


Figure 23 Create Chat Server - Advanced view

Create a server / listener home behind a router / Nat:

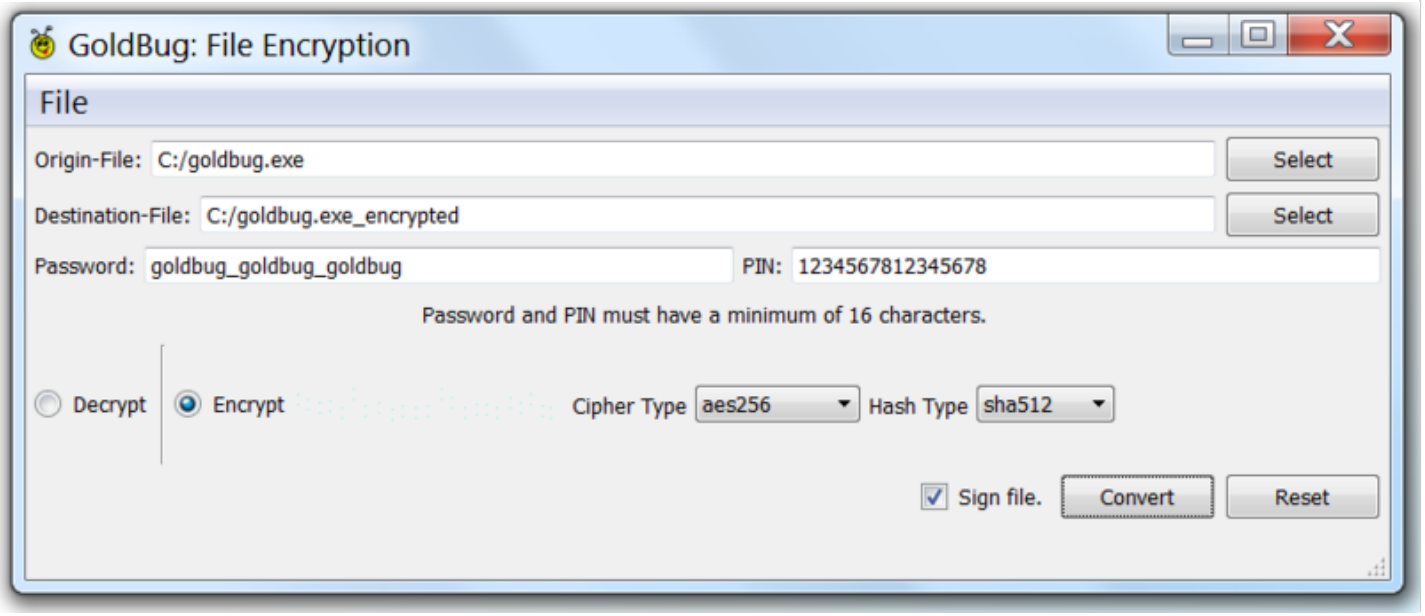
If you do not have a web server or can not find a general neighbors on the web, you can also chat server at home behind your router set up. Your friend must then not, he can directly connect as a client to your listener. But one of them must create a listener. If you want to make this behind your router / Nat home, take as geannt the machine for the listener eg 192.168.121.1 .. Then local IP address you need in your router also forward the port, that port 4710 must be forwarded by the router to 192.168.121.1: - spot-on Kernel.exe - 4710. Then, the kernel needs as well as the GoldBug.exe in your Windows Firewall be allowed. If you do everything correctly routed, the friend can connect his clients to your (external) IP address of the router (see, eg, under www.whatismyip.com) and port 4710. The important thing is that your router forwards the connection attempt from the Internet at the selected port to your local machine. This is a common and safe procedure and does not open any access to your computer, but on the port and the application is in this case as in many other programs defined that only packet be allowed in this sense. You can and must this define everything yourself and GOLDBUG does not contain code that automatically forward ports in the router, or opens or even automatically sets up a listener. Thus, it is safer and demand-oriented than other applications, configure the purposes of Nutzerfeundlichkeit themselves and this Although effort to lose weight, but also offer many ignorant people who know the technical details of port forwarding, port opening and listener-definiton, by default. So when you hear the first of them, be sure that other programs that often automatically adjust

everything and the fact that this program allows these options as manual settings by yourself, you should not put you off, give it a try and in the to trust you set technique because it blut works as described on port released, if necessary port forwarding and setting up a listener.

Tools: Encryption of files

GOLDBUG has additional tools for encryption. In the main menu, choose Tools, you can find the tool to encrypt files on your hard drive ("File Encryption Tool")

Figure 25: Tool for file encryption



GoldBug Tool: Figure 25: Encrypt Files on your Hard Disk

To be able to a file on the hard disk determine then specify the same path and select any extension or modification of the file name - then enter the password and pin (both naturally again at least 16 characters) and the radio selection buttons define whether supply the file or to be un-encrypted. Cipher- and hash type are also defined as a signature in the encryption can be included as an option in order to ensure that the encryption was done by you (or anyone else). The file encryption tool is available to replace eg potentially unsafe Truecrypt container or encrypt supplement or to backup individual files before you they transferierst - be it conventional and e-mail in GOLDBUG over Starbeam in GOLDBUG or over unsafe way - or simply to encrypt it on your hard drive or storage in online stores like Dropbox or Megaupload before.

Tools: The Rosetta CryptoPad

The tool Rosetta Crypto pad has its name from the stone of Rosette , who is in London at the Museum (see Wikipedia). He is regarded as translator for Egyptian hieroglyphs in other languages. Contained in GOLDBUG Rosetta Cryptopad consists of two dishes - as well as chat and e-mail have such own key. Swap here with a friend the Rosetta Key, give text, select the friend and whether it is encryption or decryption - and press "konverieren" button. Then the output is shown below and this you can simply print-out with the copy function and ship via conventional online communication channels such as @ -E-mail or other chat. Slow Chat by manual encoding of your chat text. It is an alternative to GnuPG (and yes it is based also on the GnuPG underlying library Libgcrypt).

Figure 26: The Rosetta CryptoPad

Release history

Figure 26:Rosetta crypto pad

Version	Date	Changes
2.7	September 26, 2015	Forward Secrecy in Email & Chat Release.
2.6	August 1, 2015	Serverless Key Share-Release.
2.5	June 19, 2015	URL-Websearch-Release.
2.1	April 20, 2015	Virtual-Keyboard-Release.
1.9	February 23, 2015	Socialist-Millionaire-Protocoll-(SMP)-Release.
1.8	January 24, 2015	E-Mail-Client-Release: Plaintext-Emails over POP3/IMAP.
1.7	December 6, 2014	Poptastic-XMAS-Release: Encrypted chat over POP3.
1.6a	November 9, 2014	2-Way-Instant-Perfect-Forward-Secrecy: "2WIPFS"-Release.
1.5	October 10, 2014	Alternative Login-Method Release
1.3	September 30, 2014	NTRU Release
1.1	September 9, 2014	Vector Update Release
1.0	September 7, 2014	File-Encryption Tool Release
0.9.09	August 20, 2014	Smiley Release
0.9.07	July 13, 2014	Adaptive Echo Release
0.9.05	May 31, 2014	Added Example Project Chat Server Release
0.9.04	April 22, 2014	SCTP & Institution Release.
0.9.02	March 13, 2014	StarBeam Analyzer Release
0.9.00	February 7, 2014	Tablet Gui Release.
0.8	December 23, 2013	Rosetta CryptoPad Release.
0.7	December 19, 2013	StarBeam Filesharing Release
0.6	October 24, 2013	El-Gamal Release
0.5	September 16, 2013	Signature-Keys Release
0.4	September 3, 2013	Kernel-Improvement Release
0.3	August 26, 2013	Geo-IP-Release
0.2	August 22, 2013	SSL-Release
0.1	July 27, 2013	based on the release of the same day of the Echo/Chat-Kernel-Servers and Application http://spot-on.sf.net, going back on another previous research project."/>

Overview of Features and further Development & Evaluation

• Spot-On is the underlaying library for the GOLDBUG Instant Messenger. • Spot-On has as well a gui and is full of adjustable options, GOLDBUG AIMS to be a desktop / mobile messenger with a smaller set of options to fit mobile or tablet devices. • Spot-On is a c ++ library as in exploratory research project investigating on encrypted communication and data transfer protocol, called the "echo protocol" or short "EMPP" protocol: Echoed Message and Presence Protocol. The package includes the Which 'libspot-on' library, is found here: spot-on.sf.net It Enables personal and group messaging, decentral p2p email, echoed IRC / Chat channels Buzz and secure file transfer with multi-encryption (SSL, RSA (PGP / GnuPGP) / ElGamal, AES, libgcrypt, OpenSSL, etc). IP Addresses are detached from Encryption Keys. It is programmed in C ++ and is the underlaying library for chat, email and messaging applications like the GOLDBUG Instant Messenger App. Spot-On can be deployed by every c-developer into chat and file sharing apps.

Short overview of Features: • Accounts: Enter your password to the account, it is not Transferred to the server, just a hash comparison is done on bothsides. • All data on your hard disk (.db files) is strong encrypted. • Gemini (end-to-end encryption key) is secured by a MAC Gemini hash. • Secure Key Transfer: Repleo encrypts your public key. • Chat over door with gold bug. • Instant Forward Secrecy with MELODICA Button: Change the encryption key end to end Whenever you want . • Set of additional password for emails (based on AES). • Send p2p emails to offline friends. • E-Signatures : Decide, if you want to send and receive emails authenticated or just non-authenticated. • Star Beam (SB) : Transmit your file into a network of encrypted packets anonymously. TCP & UDP transport for the echo protocol: UDP is ideal for VoIP echoed.

```
List of possible criteria for further evaluation
```

• Tiered application: kernel and user Interface Processes. • Use proxy capabilities? • Send email messages to offline friends? • Send email with encrypted attachments? • Having different keys for chat, email, Cryptopad, file transfer, etc.? • Is the key stuck to your IP address? • Mutual authentication access? • No hashing of a file and sending it with hash and transmitter / receiver's ID to neighbors, so it is identifyable? • Are there alternative to RSA, ElGamal like or NTRU? Can a NTRU-user chat to a RSA user? • You can use SSL or not? Selectable SSL ciphers? • Selectable hash algorithms? • Just need connectivity, no key exchange, keys are optional? • You are more autonomous? • Trust is not needed, or can be added as you define it? • Technical simplicity? • Anonymous seeds? • You can not deterministic mine, who is reading Which message (as you have no destination ID or info added)? • Free of Web-of-Trust Graphs and no mapping of connections? • Its different, its fun? • Local database stores all info in encrpyted .db 's? • Re-encode support of locally-encrypted data. • Optional authentication of messages? • You can communicate without public keys, using magnets? • Support for TCP and UDP and SCTP communications? • Support the multi-layer of encryption • Having multi encryption? eg SSL + RSA + AES? Or even ciphertext over SSL + RSA + AES (Rosetta Cryptopad ciphertext sent over encrypted channels)? • multiple listeners are possible? • A kernel is givenName? Multi-threaded ?. • IRC-like channels? • Simple IP-based firewalls? • You can define many points of connections? • Do scramblers send out fake messages ?. • You can store messages in friends? • You have the option to use to end-to-end key for communication? • You have the option to renew the end-to-end key each time you want (not only session based)? • Encrytped file transfer protocol (Starbeam)? • Using a one time magnet (OTM) for a crypto channel? • Having ipv6 support? • Having Qt 5 and up deployed? • hops are not forwarding, no routing, is it always a new wrap the message and send to just to your friend? router-less and forwarding-less protocol? • Sending a message to a friend to his dedicated connection and not to all connections? • Hiding the key exchange online? • Use several encryption keys on one file transfer? • Adding a passphrase on a file transfer? • Use it as client without a listener? • ... Over 40 criteria, Could someone analysis and write about in her / his master thesis - with different implementations in different synthesis tools Compared.

The digital encryption

of your private communication in the context of ...

Principles of the protection of private speech, communication and life: Universal Declaration of Human Rights, 1948 (Art. 12)

No one Shall be Subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. Everyone has the right to the protection of the law against interference or attacks investigated. http://www.un.org/en/documents/udhr/index.shtml#a12 http://en.wikipedia.org/wiki/Universal_Declaration_of_Human_Rights International Covenant on Civil and Political Rights, 1966 (Art. 17) 1. No one Shall be Subjected to arbitrary or unlawful interference with his privacy, family, home or correspondence, nor to unlawful attacks on his honor and reputation. 2. Everyone has the right to the protection of the law against interference or attacks investigated. http://www.ohchr.org/EN/ProfessionalInterest/Pages/CCPR.aspx http://en.wikipedia.org/wiki/International_Covenant_on_Civil_and_Political_Rights European Convention on Human Rights, 1950 (art. 8) 1.Everyone has the right to respect for his private and family life, his home and his correspondence. 2.There Shall be no interference by a public authority with the exercise of this right except as is examined in accor dance with the law and is Necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others. http://conventions.coe.int/treaty/en/Treaties/Html/005.htm http://en.wikipedia.org/wiki/European_Convention_on_Human_Rights

Charter of Fundamental Rights of the European Union, 2000 (Art. 7, 8)

Article 7. Respect for private and family life Everyone has the right to respect for his or her private and family life, home and communications. Article 8. Protection of personal data 1.Everyone has the right to the protection of personal data: concerning him or her. 2.Such data must be processed fairly for specified purposes and on the basis of the consent of the person Concerned or someother legitimate basis laid down by law. Everyone has the right of access to data has been collected Which: concerning him or her, and the right to have it rectified. 3.Compliance with synthesis rules Shall be subject to control by at independent authority. http://en.wikisource.org/wiki/Charter_of_Fundamental_Rights_of_the_European_Union http://en.wikipedia.org/wiki/Charter_of_Fundamental_Rights_of_the_European_Union

Basic Law eg for the Federal Republic of Germany 1949 (art. 2, para. 1 i. V. m.

Art. 1, para. 1)

Article 2 [Personal freedoms] (1) Every person Shall have the right to free development of his personality insofar as he does not violate the rights of others or offend against the constitutional order or the moral law. Article 1 [Human dignity - Human rights - Legally binding force of basic rights] (1) Human dignity Shall be inviolable. To respect and protect it Shall be the duty of all state authority. https://www.btg-bestellservice.de/pdf/80201000.pdf http://en.wikipedia.org/wiki/Basic_Law_for_the_Federal_Republic_of_Germany

Further: Article 1 and Article 10: Art. 1 [human dignity - human rights - Legally binding force of basic rights] (1) Human dignity Shall be inviolable. To respect and protect it Shall be the duty of all state authority. (2) The German people acknowledge inviolable and inalienable THEREFORE human rights as the basis of every community, of peace and of justice in the world. (3) The Following basic rights Shall bind the legislature, the executive and the judiciary as Directly applicable law type. 10 [Privacy of correspondence, posts and telecommunications].

Secrecy of correspondence - secrecy of telecommunications (Art. 10 para 1 of the Basic Law.) § 88 Section 1 of the secrecy of telecommunications - Telecommunications Act:

(1) The telecommunications secrecy of the content of telecommunications and their specific circumstances, in particular the fact that someone is involved in a telecommunication process or was. Telecommunications secrecy extends to the circumstances of unsuccessful connection attempts. (2) In order to maintain the secrecy of telecommunications is obliged each service provider. The duty of confidentiality continues even after the end of the activity, by which it was founded. (3) The debtor under paragraph 2, it is prohibited, or to procure another over the businesslike for the provision of telecommunications services, including the protection of their technical systems beyond what is necessary knowledge of the content or the circumstances of its telecommunications. You may knowledge of facts which are subject to the secrecy of telecommunications, use it only for the purpose referred to in clause 1. The use of such knowledge for other purposes, particularly passing to others is permissible only if this Act or any other statutory provision provides for this and explicitly refers to telecommunications operations. The obligation under § 138 of the Penal Code shall prevail. (4) the telecommunications plant is located on board an aircraft, or water, so there is a duty to maintain secrecy shall not against the person who drives the vehicle or to its delegate.

§ 206 violation of postal or telecommunications secrecy (1) Whoever without authority of another person gives a notice of facts that are subject to postal or telecommunications secrecy and have become known to him as owner or employee of a company that businesslike provides postal or telecommunications services, with imprisonment up to five years or a money penalty. (2) Likewise, anyone who illegally as owner or employee of a company referred to in paragraph 1. 1 a mission that has been entrusted to such an undertaking for the transmission and is closed, opens or gives to its contents without opening the closure under application of technical means knowledge, suppressed 2. a such an undertaking entrusted to transmit broadcast or 3. Any of the paragraph 1 or permitted in paragraph 1 or 2 referred to acts or promotes. (3) The provisions of paragraphs 1 and 2 shall apply to persons who first tasks of supervision of operations referred to in paragraph 1 companies perceive, 2nd by such organization or with its authorization to the provision of postal or telecommunications services entrusted or 3 . are engaged in the manufacture of the operation of such a company serving system or work on them. (4) Whoever without authority of another person gives a notice of facts that have become known to him to be outside the postal or telecommunications sector tätigem officials on the basis of an authorized or unauthorized interference with the postal or telecommunications secrecy, with imprisonment up to two years or with money penalty. (5) The postal secrecy the circumstances of postal traffic of certain persons as well as the contents of mail. The telecommunications secrecy, the content of telecommunications and their specific circumstances, in particular the fact that someone is involved in a telecommunication process or was. Telecommunications secrecy extends to the circumstances of unsuccessful connection attempts. http://www.gesetze-im-internet.de/gg/art_10.html http://en.wikipedia.org/wiki/Secrecy_of_correspondence http://de.wikipedia.org/wiki/Briefgeheimnis http://de.wikipedia.org/wiki/Fernmeldegeheimnis http://de.wikipedia.org/wiki/Postgeheimnis http://www.gesetze-im-internet.de/tkg_2004/__88.html http://www.gesetze-im-internet.de/stgb/__206.html

United States Constitution: Search and Seizure (Expectation of Privacy, US Supreme Court)

The right of the people to be secure in Their persons, houses, papers, and effects, against unreasonable searches and seizures, Shall not be violated, and no Warrants Shall issue, but upon probable cause, supported by Oath or affirmation, and particularly Describing the place to be searched, and the persons or things to be seized. http://www.usconstitution.net/const.html

Web Page

More information can be found on the website:

http://goldbug.sf.net

Retrieved from "https://en.wikibooks.org/w/index.php?title=Goldbug&oldid=2997807"
